Application Techniques

Allen-Bradley

Controller Examples for EtherNet/IP Network Communication with PowerFlex 750-Series Drives

PLC-5, SLC 500, and MicroLogix 1100/1400 Controllers

Important User Information

Solid-state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation® sales office or online at http://www.rockwellautomation.com/literature/) describes some important differences between solid-state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid-state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

	**WARNING:** Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
	**ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.
	**SHOCK HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
	**BURN HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

Logic Command/Status Words: PowerFlex 750-Series Drives

This manual provides information about how to use Rockwell Automation PLC-5®, SLC™ 500, and MicroLogix™ 1100/1400 controllers with PowerFlex® 750-Series drives over an EtherNet/IP network.

Conventions Used in This Manual

The following conventions are used throughout this manual:

- Parameter names are shown in the format 'Device' **Parameter xx - [*]** or 'Host' **Parameter xx - [*]**. The xx represents the parameter number. The * represents the parameter name—for example, 'Device' **Parameter 01 - [Operating Mode]**.
- The firmware revision number (FRN) is displayed as FRN *X.xxx*, where '*X*' is the major revision number and '*xxx*' is the minor revision number.
- For the screen captures in this manual, the following software was used:
  - RSLinx® Classic software, version 2.52
  - RSLogix™ 5 software, version 7.20
  - RSLogix 500 software, version 7.20

  Different versions of the software may differ in appearance and procedures.

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 sales and support offices, over 500 authorized distributors, and over 250 authorized systems integrators located through the United States alone. In addition, Rockwell Automation representatives are in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for the following:

- Sales and order support
- Product technical training
- Warranty support
- Support service agreements

Technical Product Assistance

For technical assistance, please access the Allen-Bradley Technical Support website at http://www.ab.com/support/abdrives or contact Rockwell Automation.

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
Network Communication Option Module Installation Instructions, publication 750COM-IN002	Information on the installation of PowerFlex 750-Series Network Communication Modules.
EtherNet/IP Media Planning and Installation Manual, ODVA publication 148 (1)	Information on the planning, installation, and techniques used to implement an EtherNet/IP network.
EtherNet/IP Network Infrastructure Guidelines, ODVA publication 35(1)	
Ethernet Design Considerations Reference Manual, publication ENET-RM002	
EtherNet/IP Embedded Switch Technology - Linear and Device-level Ring Topologies, publication ENET-AP005	
DriveExplorer website http://www.ab.com/drives/driveexplorer, and online help (2)	Information on using the DriveExplorer™ software tool.
DriveExecutive website http://www.ab.com/drives/drivetools, and online help (2)	Information on using the DriveExecutive™ software tool.
PowerFlex 750-Series Drive Installation Instructions, publication 750-IN001	Information on installing, programming, and technical data of PowerFlex 750-Series drives.
PowerFlex 750-Series Drive Programming Manual, publication 750-PM001	
PowerFlex 750-Series Drive Technical Data, publication 750-TD001	
PowerFlex 20-HIM-A6/-C6S HIM (Human Interface Module) User Manual, publication 20HIM-UM001	Information on the installation and use of PowerFlex 20-HIM-A6 or 20-HIM-C6S HIMs.
Getting Results with RSLinx Guide, publication LINX-GR001 and online help (2)	Information on using RSLinx Classic software.
RSLogix Emulate 5/500 Getting Results Guide, publication EMULAT-GR002 and online help (2)	Information on how to install and navigate the RSLogix Emulate software for ladder logic programming with Allen-Bradley® PLC-5 and SLC 500 processors.
RSLogix 500 Getting Results Guide, publication LG500-GR002 and online help (2)	Information on using the RSLogix 500 software tool.
Enhanced and Ethernet PLC-5 Programmable Controllers User Manual, publication 1785-UM012	Information to help design, operate and maintain an Enhanced and Ethernet PLC-5 programmable controller system.
SLC 500 Modular Hardware Style User Manual, publication 1747-UM011	Information on how to install, wire, use, and troubleshoot the SLC 500 controller with 1747-L5-xxx module.
MicroLogix 1100 Programmable Controllers User Manual, publication 1763-UM001	Information to install, wire, and troubleshoot the MicroLogix 1100 controller.
MicroLogix 1400 Programmable Controllers User Manual, publication 1766-UM001	Information to install, wire, and troubleshoot the MicroLogix 1400 controller.

(1) Use this link to the ODVA EtherNet/IP library: http://odva.org/Home/ODVATECHNOLOGIES/EtherNetIP/EtherNetIPLibrary/tabid/76/Default.aspx.
(2) The online help is installed with the software.

You can view or download publications at http://www.rockwellautomation.com/literature. To order paper copies of technical documentation, contact your local Allen-Bradley distributor or Rockwell Automation sales representative.

To find your local Rockwell Automation distributor or sales representative, visit http://www.rockwellautomation.com/locations.

For information, such as firmware updates or answers to drive-related questions, go to the Drives Service & Support website at http://www.ab.com/support/abdrives and click the Downloads or Knowledgebase link.

Configuring the I/O

This chapter provides instructions on how to configure a Rockwell Automation PLC-5, SLC 500, or MicroLogix 1100/1400 controller to communicate over an EtherNet/IP network with PowerFlex 750-Series drives.

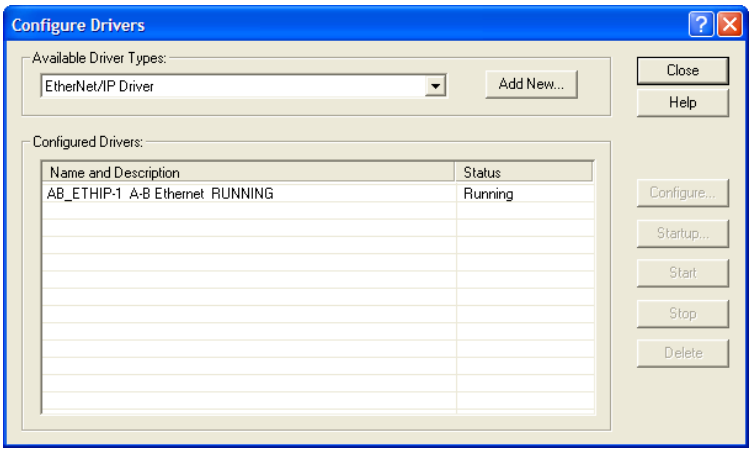
Using RSLinx Classic Software

RSLinx Classic software, in all its variations (Lite, Gateway, OEM, and so forth), is used to provide a communication link between the computer, network, and controller. RSLinx Classic software requires its network-specific driver to be configured before communication is established with network devices. Follow these steps to configure the RSLinx driver.

1. Start RSLinx Classic software.
2. From the Communications menu, choose Configure Drivers to display the Configure Drivers dialog box.
3. From the Available Driver Types pull-down menu, choose EtherNet/IP Driver.
4. Click Add New to display the Add New RSLinx Driver dialog box.
5. Use the default name or type a name.
6. Click OK.

   The ‘Configure driver:’ dialog box appears.
7. Depending on your application, select either the browse local or remote subnet option.
8. Click OK.

The Configure Drivers dialog box reappears with the new driver in the Configured Drivers list.

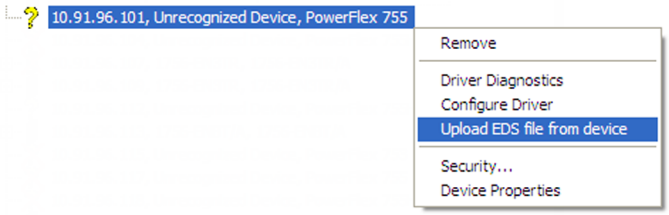


9. Click Close to close the Configure Drivers dialog box.
10. Keep RSLinx software running and verify that your computer recognizes the drive.
    a. From the Communications menu, choose RSWho.
    b. In the menu tree, click '+' next to the Ethernet driver.

    Note that two other RSLinx drivers (Ethernet devices or Remote Devices via Linx Gateway) may be used. Use one of these drivers if the 'EtherNet/IP Driver' cannot see your drive.

Uploading the Electronic Data Sheet (EDS) File

The option module has embedded EDS files for PowerFlex 750-Series drives. This eliminates the need to download an EDS file from the Rockwell Automation website. Instead, RSLinx software is used to upload the EDS file from the option module. To do this, right-click the EtherNet/IP drive that is shown as a yellow question mark or 'Unrecognized Device' in the RSWho dialog box and choose 'Upload EDS file from device' as shown below.



Limitations Using a PLC-5, SLC 500, or MicroLogix 1100/ 1400 Controller

Controlling I/O with explicit messages is relatively complex compared to normal implicit I/O control.

ControlLogix® and CompactLogix™ controllers with EtherNet/IP provide the easiest and most integrated form of implicit I/O control for a PowerFlex drive. RSLogix 5000 software, version 16.00 or later, for ControlLogix and CompactLogix controllers contains integrated profiles for PowerFlex drives that, with a few clicks of the mouse, automatically create all controller tags and an implicit connection at the specified Requested Packet Interval to control the drive. This connection is monitored at both ends to verify that the controller and drive are communicating. A watchdog will cause a drive fault if the drive does not respond within approximately 100 milliseconds. Therefore, using a ControlLogix or CompactLogix controller is by far the much preferred method of controlling drives on EtherNet/IP.

If you are not using either of these type of controllers, then PowerFlex drives on EtherNet/IP can be controlled with explicit messages using PLC-5, SLC 500 or MicroLogix 1100/1400 controllers with the following limitations:

- An explicit message is a much slower form of control and is non-deterministic. This means that you cannot guarantee how long the drive will take to start up or stop when the command is given. Therefore, all equipment used in this manner should be subject to a risk assessment, taking into account the mechanical and electrical implementation.
- A timeout value (in seconds) in the EtherNet/IP option module will issue a drive fault if a message is not received from the controller within the specified time. However, the controller has no way of detecting a loss of communication to the drive until the next cycle of explicit messages. This is another factor in the risk assessment.
- Any additional drives to be controlled will require additional explicit messages for their control, and they need to be carefully sequenced. Most controllers have small communication queues (see its User Manual), which need to be carefully managed if messages are not to be lost.
- Each controller has a limited number of communication connections (see its User Manual for maximum connections), which will limit the number of drives that can be connected.

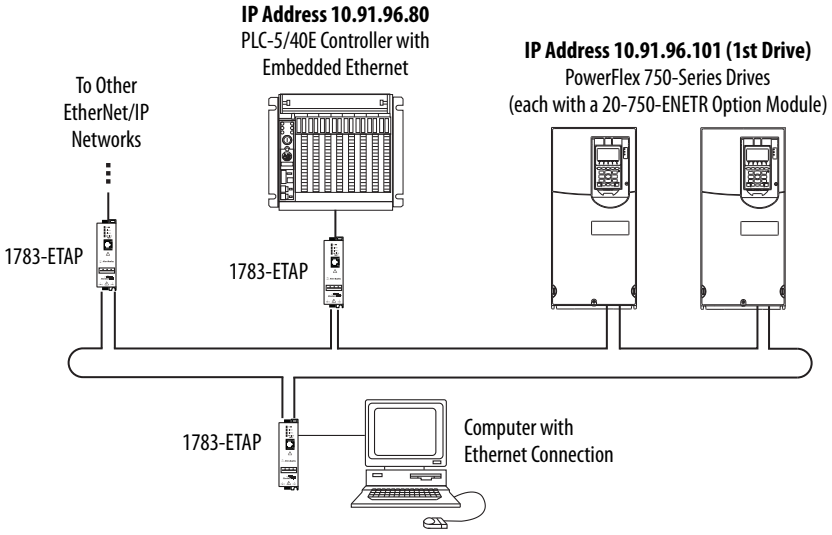
In summary, unlike a ControlLogix or CompactLogix controller, programming a PLC-5, SLC 500 or MicroLogix 1100/1400 controller by using RSLogix 5 or RSLogix 500 software with explicit messages is more difficult, and produces a more complex program.

PLC-5 Controller Example

IMPORTANT The PLC-5 must be Series E (Rev. D.1 or later) to support the MultiHop feature that routes messaging to the drive.

After the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive is configured, the connected drive and option module or adapter will be a single node on the network. This section provides the steps needed to configure a simple DLR topology EtherNet/IP network (see Figure 1). In our example, we will configure a PLC-5/40E controller to communicate with the first drive in the network ring using Logic Command/Status, Reference/Feedback, and 32 Datalinks (16 to read data and 16 to write data) over the network.

Figure 1 - Example PLC-5 Controller EtherNet/IP Device-level Ring Network



Configuring Parameters for Network I/O

Because the I/O for the drive is message-based, there is no need to configure any I/O inside the RSLogix 5 project, version 7.00 or later, until using the I/O as described in Chapter 2.

However, to get the option module or embedded adapter to operate with the I/O created in Chapter 2, you need to configure the option module to accept the I/O and the drive to point to the appropriate Datalinks.

Because the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive always uses the 32-bit Logic Status and 32-bit Feedback, at least two 32-bit words must be accounted for in the controller input image. The option module or embedded adapter also uses the 32-bit Logic Command and 32-bit Reference, requiring at least two 32-bit words that must be accounted for in the controller output image.

If any or all of the drive's sixteen 32-bit Datalinks are used the following applies:

- An additional 32-bit word for 'each' [**DL From Net xx**] Datalink used to write to drive or peripheral parameters must be accounted for in the controller input image.
- An additional 32-bit word for 'each' [**DL To Net xx**] Datalink used to read data must be accounted for in the controller output image.

Option Module or Embedded Adapter Parameter Settings for PLC-5 Controller Example

These option module or embedded adapter settings were used for the example ladder logic program in this section.

Option Module 'Host' Parameter or Embedded Adapter Parameter	Value	Description
01 - [DL From Net 01]	370	Points to drive Par. 370 - [Stop Mode A]
02 - [DL From Net 02]	371	Points to drive Par. 371 - [Stop Mode B]
03 - [DL From Net 03]	535	Points to drive Par. 535 - [Accel Time 1]
04 - [DL From Net 04]	536	Points to drive Par. 536 - [Accel Time 2]
05 - [DL From Net 05]	537	Points to drive Par. 537 - [Decel Time 1]
06 - [DL From Net 06]	538	Points to drive Par. 538 - [Decel Time 2]
07 - [DL From Net 07]	539	Points to drive Par. 539 - [Jog Acc Dec Time]
08 - [DL From Net 08]	556	Points to drive Par. 556 - [Jog Speed 1]
09 - [DL From Net 09]	557	Points to drive Par. 557 - [Jog Speed 2]
10 - [DL From Net 10]	571	Points to drive Par. 571 - [Preset Speed 1]
11 - [DL From Net 11]	572	Points to drive Par. 572 - [Preset Speed 2]
12 - [DL From Net 12]	573	Points to drive Par. 573 - [Preset Speed 3]
13 - [DL From Net 13]	574	Points to drive Par. 574 - [Preset Speed 4]
14 - [DL From Net 14]	575	Points to drive Par. 575 - [Preset Speed 5]
15 - [DL From Net 15]	576	Points to drive Par. 576 - [Preset Speed 6]
16 - [DL From Net 16]	577	Points to drive Par. 577 - [Preset Speed 7]
17 - [DL To Net 01]	370	Points to drive Par. 370 - [Stop Mode A]
18 - [DL To Net 02]	371	Points to drive Par. 371 - [Stop Mode B]
19 - [DL To Net 03]	535	Points to drive Par. 535 - [Accel Time 1]
20 - [DL To Net 04]	536	Points to drive Par. 536 - [Accel Time 2]
21 - [DL To Net 05]	537	Points to drive Par. 537 - [Decel Time 1]
22 - [DL To Net 06]	538	Points to drive Par. 538 - [Decel Time 2]
23 - [DL To Net 07]	539	Points to drive Par. 539 - [Jog Acc Dec Time]
24 - [DL To Net 08]	556	Points to drive Par. 556 - [Jog Speed 1]
25 - [DL To Net 09]	557	Points to drive Par. 557 - [Jog Speed 2]
26 - [DL To Net 10]	571	Points to drive Par. 571 - [Preset Speed 1]
27 - [DL To Net 11]	572	Points to drive Par. 572 - [Preset Speed 2]
28 - [DL To Net 12]	573	Points to drive Par. 573 - [Preset Speed 3]
29 - [DL To Net 13]	574	Points to drive Par. 574 - [Preset Speed 4]
30 - [DL To Net 14]	575	Points to drive Par. 575 - [Preset Speed 5]
31 - [DL To Net 15]	576	Points to drive Par. 576 - [Preset Speed 6]
32 - [DL To Net 16]	577	Points to drive Par. 577 - [Preset Speed 7]

TIP The **[DL From Net xx]** parameters are inputs into the drive that come from controller outputs (for example, data to write to a drive parameter). The **[DL To Net xx]** parameters are outputs from the drive that go to controller inputs (for example, data to read a drive parameter).

Creating RSLogix 5 Project, Version 7.00 or Later

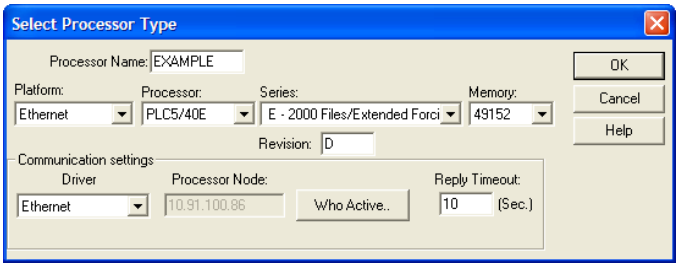
To transmit (read and write) data between the controller and drive, you must create message instructions that allocate data table addresses in the controller for Logic Command/Status, Reference/Feedback, and Datalinks. Note that three messages need to be configured. The timeout message has to be executed first before the Logic Command, Reference, and DL To/From Net Datalink messages will work. For more information on N42:3 and N45 target device data table addresses, see N-Files on page 89.

Select the Controller

1. Start RSLogix 5 software.

   The RSLogix 5 dialog box appears.

2. From the File menu, choose New to display the Select Processor Type dialog box.



3. Assign a name for the processor.
4. From the pull-down menus, choose the appropriate selections to match your PLC-5 controller and application.
5. Click OK.

   The RSLogix 5 project dialog box appears.

Create PLC-5 Ladder Logic for the Control Timeout

1. In the RSLogix 5 project dialog box treeview under Program Files double-click LAD 2.
2. Insert a ladder rung.
3. Double-click the rung to display the rung editor.
4. Enter **MSG MGxx:n**, where:

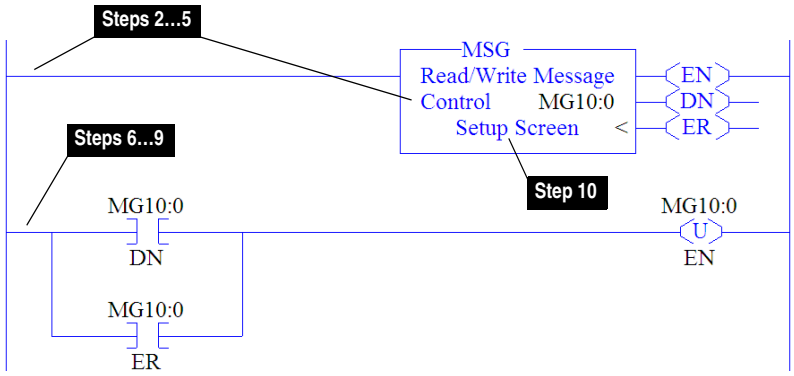
   xx is an unused data file number (for example, MG10:n), and
   n is an unused element of the data file chosen for xx (for example, MG10:0)
5. Press Enter.
6. Insert another separate rung.
7. Double-click the rung to display the rung editor.
8. Enter **BST XIC MGxx:n/DN NXB XIC MGxx:n/ER BND OTU MGxx:n/EN**, where:

   xx and n must correspond to the assigned data file number and element (for example, MG10:0) for the message created in steps 2…5.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

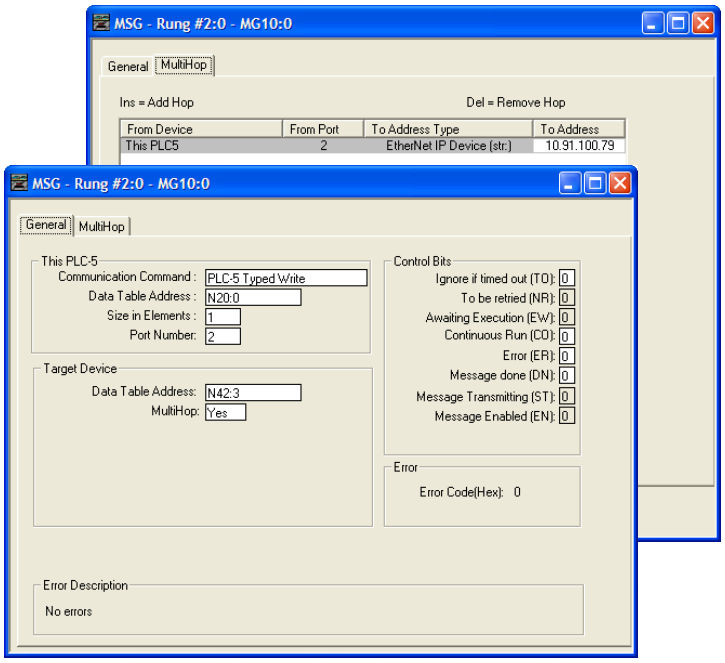
9. Press Enter.
10. In the MSG instruction (Figure 2), double-click Setup Screen to launch the message configuration dialog box (Figure 3).

Figure 2 - PLC-5 Ladder Logic for the Control Timeout



11. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 3 - PLC-5 Message Configuration Dialog Boxes for the Control Timeout



General Tab: This PLC-5	Setting
Communication Command	**PLC-5 Typed Write**. The controller type and command type for the controller to write the control timeout value to the drive.
Data Table Address (1)	**N20:0**. An unused controller data table address containing the control timeout value to be written.
Size in Elements (2)	**1**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
Port Number	**2**. Controller port to which the network is connected.
General Tab: Target Device	**Setting**
Data Table Address (3)	**N42:3**. Specific starting address of the destination file in the drive.
MultiHop	**Yes**. Enables communication to allow network messaging to be routed to the option module/drive. When 'Yes' is selected, a MultiHop tab appears on the message configuration dialog box.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.

(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.

(3) For details on setting the control timeout value and its function, see N-Files on page 89. The Control Timeout (N42:3) is stored in RAM. If the EtherNet/IP option module is power cycled, the Control Timeout Message must be re-sent. If the Control Timeout is not changed from a non-zero value, the control message (page 17) will error out.

TIP The Control Timeout (N42:3) must be changed to a non-zero value (5...20 seconds recommended). If the Control Timeout is not changed from a non-zero value, the control message (page 17) will error out. The Control Timeout is stored in RAM. If the option module is power cycled, the Control Timeout Message must be re-sent.

Create PLC-5 Ladder Logic for the Logic Status, Feedback, and DL from Net Datalinks

1. Insert another separate rung.
2. Double-click the rung to display the rung editor.
3. Enter **MSG MGxx:n**, where:

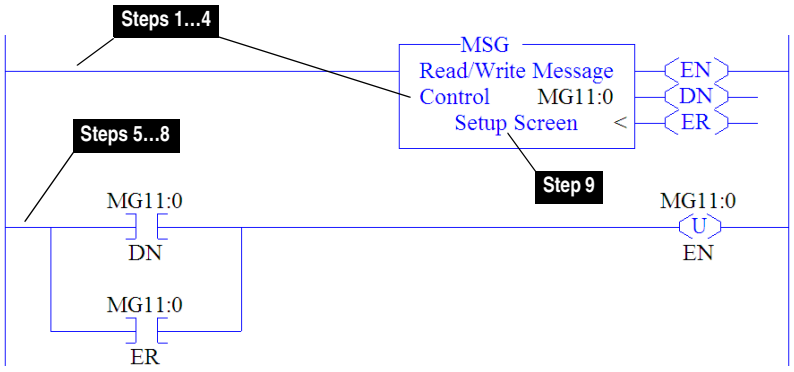
   xx is an unused data file number (for example, MG11:n), and
   n is an unused element of the data file chosen for xx (for example, MG11:0)
4. Press Enter.
5. Insert another separate rung.
6. Double-click the rung to display the rung editor.
7. Enter **BST XIC MGxx:n/DN NXB XIC MGxx:n/ER BND OTU MGxx:n/EN**, where:

   xx and n must correspond to the assigned data file number and element (for example, MG11:0) for the message created in steps 1…4.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

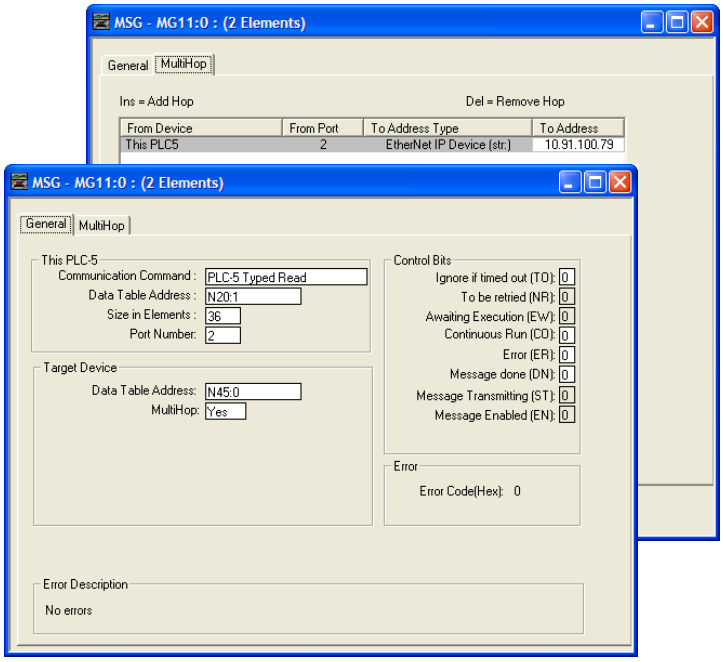
8. Press Enter.
9. In the MSG instruction (Figure 4), double-click Setup Screen to launch the message configuration dialog box (Figure 5).

Figure 4 - PLC-5 Ladder Logic for the Logic Status, Feedback, and DL from Net Datalinks



10. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 5 - PLC-5 Message Configuration Dialog Boxes for the Logic Status Feedback, and DL from Net Datalinks



General Tab: This PLC-5	Setting
Communication Command	**PLC-5 Typed Read**. The controller type and command type for the controller to read data from the drive.
Data Table Address (1)	**N20:1**. An unused controller data table address containing the data to be read from the drive.
Size in Elements (2)	**36**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
Port Number	**2**. Controller port to which the network is connected.
General Tab: Target Device	**Setting**
Data Table Address (3)	**N45:0**. Specific starting address of the source file in the drive.
MultiHop	**Yes**. Enables communication to allow network messaging to be routed to the option module/drive. When 'Yes' is selected, a MultiHop tab appears on the message configuration dialog box.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.
(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.
(3) For N-File details, see N-Files on page 89.

Create PLC-5 Ladder Logic for the Logic Command, Reference, and DL to Net Datalinks

1. Insert another separate rung.
2. Double-click the rung to display the rung editor.
3. Enter **MSG MGxx:n**, where:

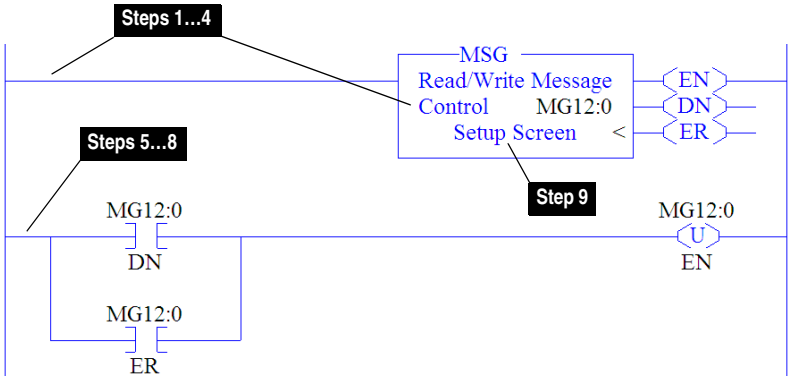
   xx is an unused data file number (for example, MG12:n), and
   n is an unused element of the data file chosen for xx (for example, MG12:0)
4. Press Enter.
5. Insert another separate rung.
6. Double-click the rung to display the rung editor.
7. Enter **BST XIC MGxx:n/DN NXB XIC MGxx:n/ER BND OTU MGxx:n/EN**, where:

   xx and n must correspond to the assigned data file number and element (for example, MG12:0) for the message created in steps 1…4.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

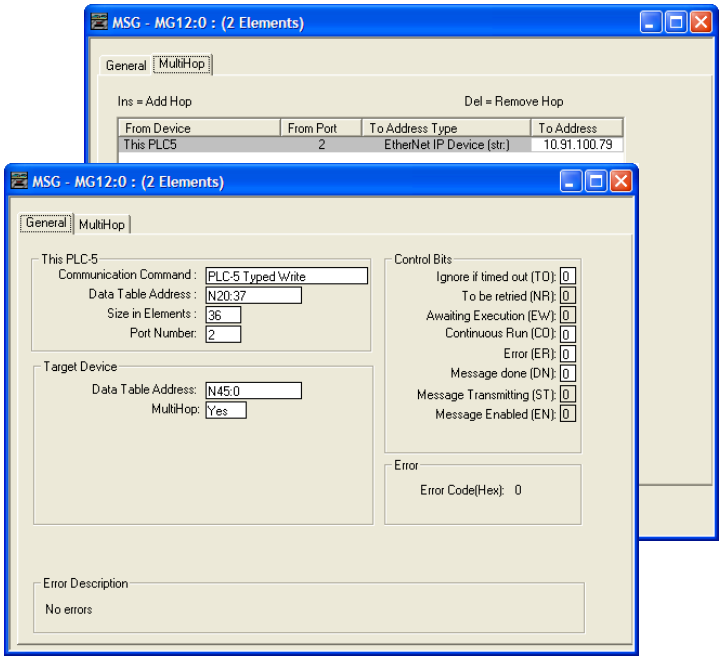
8. Press Enter.
9. In the MSG instruction (Figure 6), double-click Setup Screen to launch the message configuration dialog box (Figure 7).

Figure 6 - PLC-5 Ladder Logic for the Logic Command, Reference, and DL to Net Datalinks



10. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 7 - PLC-5 Message Configuration Dialog Boxes for the Logic Command, Reference, and DL to Net Datalinks



General Tab: This PLC-5	Setting
Communication Command	**PLC-5 Typed Write**. The controller type and command type for the controller to write data to the drive.
Data Table Address (1)	**N20:37**. An unused controller data table address containing the data to be written to the drive.
Size in Elements (2)	**36**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
Port Number	**2**. Controller port to which the network is connected.
General Tab: Target Device	**Setting**
Data Table Address (3)	**N45:0**. Specific starting address of the destination file in the drive.
MultiHop	**Yes**. Enables communication to allow network messaging to be routed to the option module/drive. When 'Yes' is selected, a MultiHop tab appears on the message configuration dialog box.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.
(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.
(3) For N-File details, see N-Files on page 89.

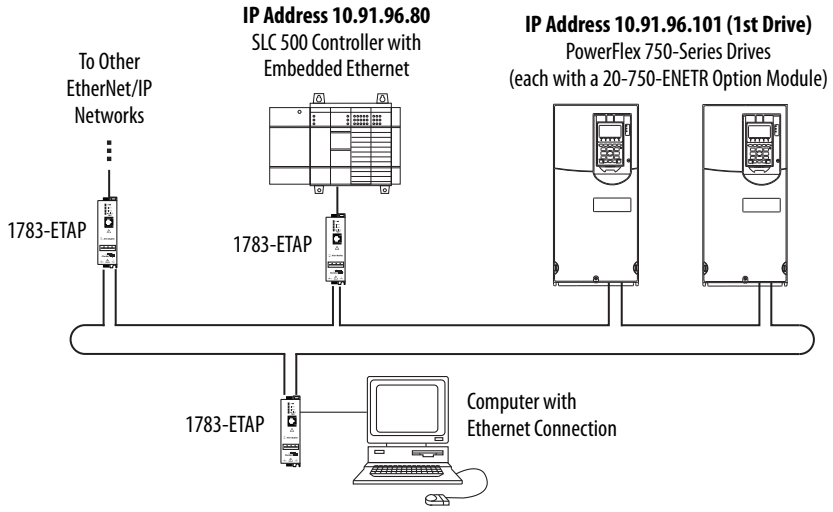
TIP This message will error out if the Control Timeout value is not changed from a non-zero value. See page 13 for writing a value to the Control Timeout.

TIP If the controller is controlling more than one drive, it is recommended to intersperse the control I/O messaging for each drive to conserve network bandwidth and decrease response time. That is, sequence the message instructions for each drive so that its group of messages will occur at a different time than those for another drive.

SLC 500 Controller Example

After the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive is configured, the connected drive and option module or adapter will be a single node on the network. This section provides the steps needed to configure a simple DLR topology EtherNet/IP network (see Figure 8). In our example, we will configure a SLC 500 controller to communicate with the first drive in the network ring using Logic Command/Status, Reference/Feedback, and 32 Datalinks (16 to read data and 16 to write data) over the network.

Figure 8 - Example SLC 500 Controller EtherNet/IP Device-level Ring Network



Configuring Parameters for Network I/O

Because the I/O for the drive is message-based, there is no need to configure any I/O inside the RSLogix 500 project, version 7.00 or later, until using the I/O as described in Chapter 2.

However, to get the option module or embedded adapter to operate with the I/O created in Chapter 2, we need to configure the option module to accept the I/O and drive to point to the appropriate Datalinks.

Because the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive always uses the 32-bit Logic Status and 32-bit Feedback, at least two 32-bit words must be accounted for in the controller input image. The option module or embedded adapter also uses the 32-bit Logic Command and 32-bit Reference, requiring at least two 32-bit words that must be accounted for in the controller output image. If any or all of the drive's sixteen 32-bit Datalinks are used the following applies:

- An additional 32-bit word for 'each' [**DL From Net xx**] Datalink used to write to drive or peripheral parameters must be accounted for in the controller input image.
- An additional 32-bit word for 'each' [**DL To Net xx**] Datalink used to read data must be accounted for in the controller output image.

Option Module or Embedded Adapter Parameter Settings for SLC 500 Controller Example

These option module or embedded adapter settings were used for the example ladder logic program in this section.

Option Module *Host* Parameter or Embedded Adapter Parameter	Value	Description
01 - [DL From Net 01]	370	Points to drive Par. 370 - [Stop Mode A]
02 - [DL From Net 02]	371	Points to drive Par. 371 - [Stop Mode B]
03 - [DL From Net 03]	535	Points to drive Par. 535 - [Accel Time 1]
04 - [DL From Net 04]	536	Points to drive Par. 536 - [Accel Time 2]
05 - [DL From Net 05]	537	Points to drive Par. 537 - [Decel Time 1]
06 - [DL From Net 06]	538	Points to drive Par. 538 - [Decel Time 2]
07 - [DL From Net 07]	539	Points to drive Par. 539 - [Jog Acc Dec Time]
08 - [DL From Net 08]	556	Points to drive Par. 556 - [Jog Speed 1]
09 - [DL From Net 09]	557	Points to drive Par. 557 - [Jog Speed 2]
10 - [DL From Net 10]	571	Points to drive Par. 571 - [Preset Speed 1]
11 - [DL From Net 11]	572	Points to drive Par. 572 - [Preset Speed 2]
12 - [DL From Net 12]	573	Points to drive Par. 573 - [Preset Speed 3]
13 - [DL From Net 13]	574	Points to drive Par. 574 - [Preset Speed 4]
14 - [DL From Net 14]	575	Points to drive Par. 575 - [Preset Speed 5]
15 - [DL From Net 15]	576	Points to drive Par. 576 - [Preset Speed 6]
16 - [DL From Net 16]	577	Points to drive Par. 577 - [Preset Speed 7]
17 - [DL To Net 01]	370	Points to drive Par. 370 - [Stop Mode A]
18 - [DL To Net 02]	371	Points to drive Par. 371 - [Stop Mode B]
19 - [DL To Net 03]	535	Points to drive Par. 535 - [Accel Time 1]
20 - [DL To Net 04]	536	Points to drive Par. 536 - [Accel Time 2]
21 - [DL To Net 05]	537	Points to drive Par. 537 - [Decel Time 1]
22 - [DL To Net 06]	538	Points to drive Par. 538 - [Decel Time 2]
23 - [DL To Net 07]	539	Points to drive Par. 539 - [Jog Acc Dec Time]
24 - [DL To Net 08]	556	Points to drive Par. 556 - [Jog Speed 1]
25 - [DL To Net 09]	557	Points to drive Par. 557 - [Jog Speed 2]
26 - [DL To Net 10]	571	Points to drive Par. 571 - [Preset Speed 1]
27 - [DL To Net 11]	572	Points to drive Par. 572 - [Preset Speed 2]
28 - [DL To Net 12]	573	Points to drive Par. 573 - [Preset Speed 3]
29 - [DL To Net 13]	574	Points to drive Par. 574 - [Preset Speed 4]
30 - [DL To Net 14]	575	Points to drive Par. 575 - [Preset Speed 5]
31 - [DL To Net 15]	576	Points to drive Par. 576 - [Preset Speed 6]
32 - [DL To Net 16]	577	Points to drive Par. 577 - [Preset Speed 7]

TIP The **[DL From Net xx]** parameters are inputs into the drive that come from controller outputs (for example, data to write to a drive parameter). The **[DL To Net xx]** parameters are outputs from the drive that go to controller inputs (for example, data to read a drive parameter).

Creating RSLogix 500 Project, Version 7.00 or Later

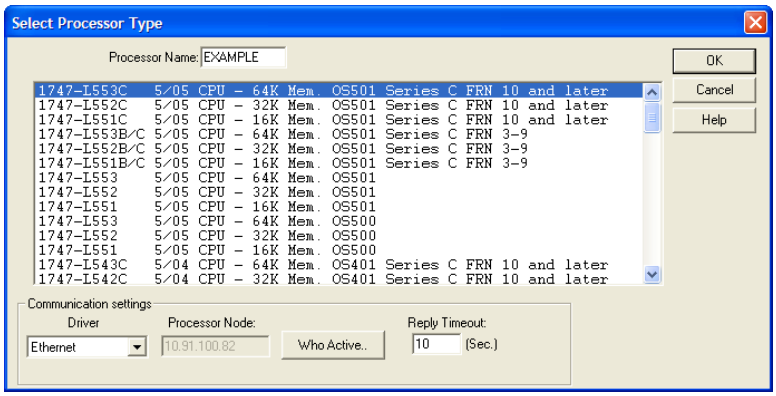
To transmit (read and write) data between the controller and drive, you must create message instructions that allocate data table addresses in the controller for Logic Command/Status, Reference/Feedback, and Datalinks. Note that three messages need to be configured. The timeout message has to be executed first before the Logic Command, Reference, and DL To/From Net Datalink messages will work. For more information on N42:3 and N45 target device data table addresses, see N-Files on page 89.

Select the Controller

1. Start RSLogix 500 software.

   The RSLogix 500 dialog box appears.

2. From the File menu, choose New to display the Select Processor Type dialog box.



3. Assign a name for the processor.
4. In the list, select a 1747-L55x type controller.
5. Choose the appropriate choices for the fields in the dialog box to match your application.
6. Click OK.

   The RSLogix 500 project dialog box appears.

Create SLC 500 Ladder Logic for the Control Timeout

1. In the RSLogix 500 project dialog box treeview under Program Files double-click LAD 2.
2. Insert a ladder rung.
3. Double-click the rung to display the rung editor.
4. Enter **MSG WRITE 500CPU LOCAL Nxx:n**, where:

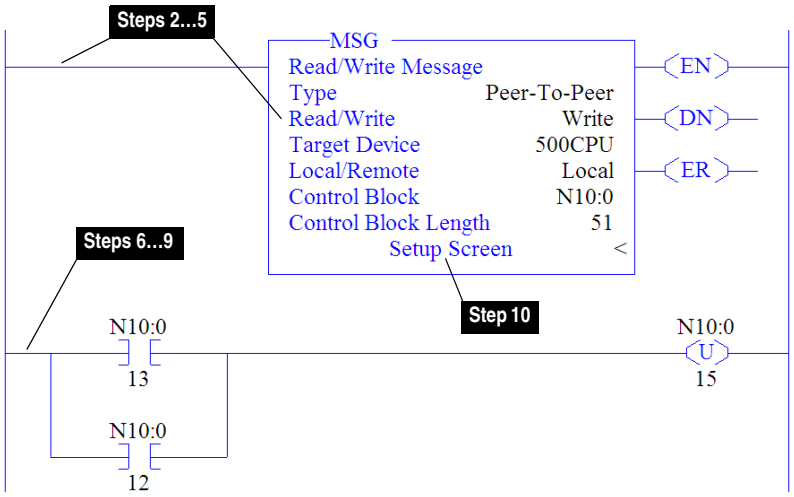
   xx is an unused data file number (for example, N10:n), and
   n is an unused element of the data file chosen for xx (for example, N10:0)
5. Press Enter.
6. Insert another separate rung.
7. Double-click the rung to display the rung editor.
8. Enter **BST XIC Nxx:n/13 NXB XIC Nxx:n/12 BND OTU Nxx:n/15**, where:

   xx and n must correspond to the assigned data file number and element (for example, N10:0) for the message created in steps 2…5.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

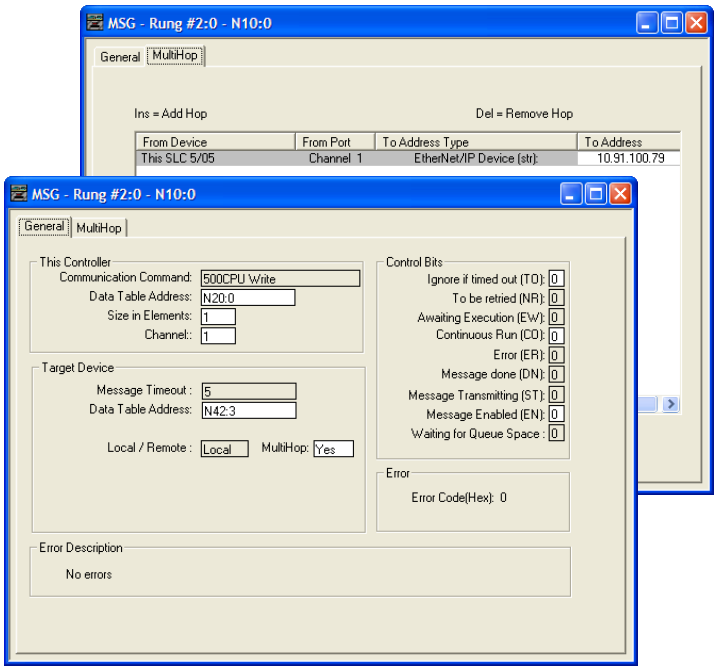
9. Press Enter.
10. In the MSG instruction (Figure 9), double-click Setup Screen to launch the message configuration dialog box (Figure 10).

Figure 9 - SLC 500 Ladder Logic for the Control Timeout



11. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 10 - SLC 500 Message Configuration Dialog Boxes for the Control Timeout



General Tab: This Controller	Setting
Communication Command	This setting is dimmed (unavailable) and is established when the message is created in the ladder rung.
Data Table Address (1)	**N20:0**. An unused controller data table address containing the control timeout value to be written.
Size in Elements (2)	**1**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
Channel	**1**. Controller port to which the network is connected.
General Tab: Target Device	**Setting**
Message Timeout	This setting is dimmed (unavailable). Message timeout duration in seconds.
Data Table Address (3)	**N42:3**. Specific starting address of the destination file in the drive.
MultiHop	**Yes**. Enables communication to allow network messaging to be routed to the option module/drive. When 'Yes' is selected, a MultiHop tab appears on the message configuration dialog box.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.
(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.
(3) For details on setting the control timeout value and its function, see N-Files on page 89. The Control Timeout (N42:3) is stored in RAM. If the EtherNet/IP option module is power cycled, the Control Timeout Message must be re-sent. If the Control Timeout is not changed from a non-zero value, the control message (page 26) will error out.

TIP The Control Timeout (N42:3) must be changed to a non-zero value (5...20 seconds recommended). If the Control Timeout is not changed from a non-zero value, the control message (page 26) will error out. The Control Timeout is stored in RAM. If the option module is power cycled, the Control Timeout Message must be re-sent.

Create SLC 500 Ladder Logic for the Logic Status, Feedback, and DL from Net Datalinks

1. Insert another separate rung.
2. Double-click the rung to display the rung editor.
3. Enter **MSG READ 500CPU LOCAL Nxx:n**, where:

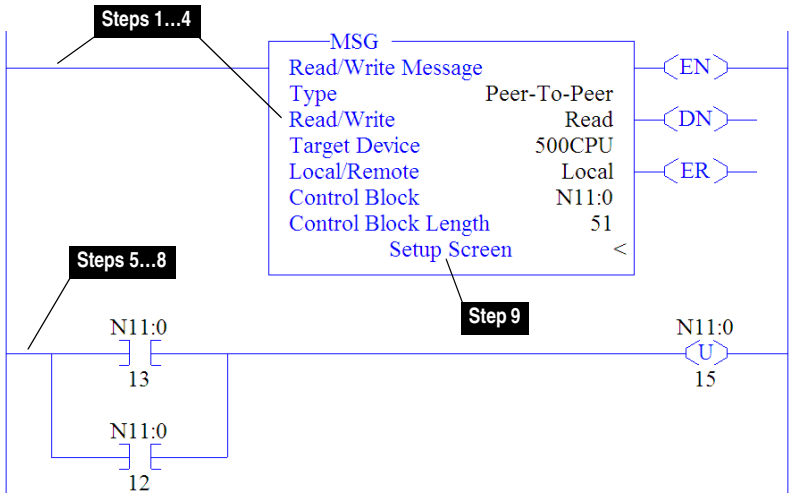
   xx is an unused data file number (for example, N11:n), and
   n is an unused element of the data file chosen for xx (for example, N11:0)
4. Press Enter.
5. Insert another separate rung.
6. Double-click the rung to display the rung editor.
7. Enter **BST XIC Nxx:n/13 NXB XIC Nxx:n/12 BND OTU Nxx:n/15**, where:

   xx and n must correspond to the assigned data file number and element (for example, N11:0) for the message created in steps 1…4.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

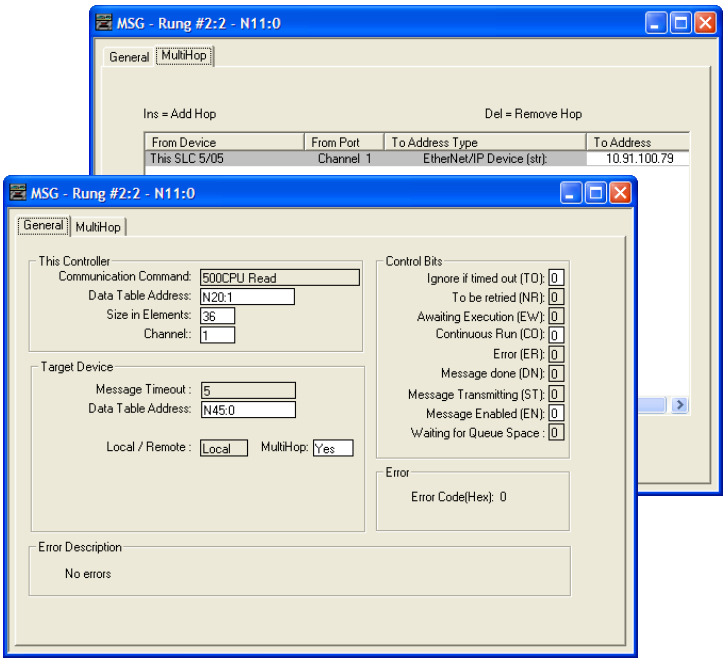
8. Press Enter.
9. In the MSG instruction (Figure 11), double-click Setup Screen to launch the message configuration dialog box (Figure 12).

Figure 11 - SLC 500 Ladder Logic for the Logic Status, Feedback, and DL from Net Datalinks



10. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 12 - SLC 500 Message Configuration Dialog Boxes for the Logic Status, Feedback, and DL from Net Datalinks



General Tab: This Controller	Setting
Communication Command	This setting is dimmed (unavailable) and is established when the message is created in the ladder rung.
Data Table Address (1)	**N20:1**. An unused controller data table address containing the data to be read from the drive.
Size in Elements (2)	**36**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
Channel	**1**. Controller port to which the network is connected.
General Tab: Target Device	**Setting**
Message Timeout	This setting is dimmed (unavailable). Message timeout duration in seconds.
Data Table Address (3)	**N45:0**. Specific starting address of the source file in the drive.
MultiHop	**Yes**. Enables communication to allow network messaging to be routed to the option module/drive. When 'Yes' is selected, a MultiHop tab appears on the message configuration dialog box.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.
(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.
(3) For N-File details, see N-Files on page 89.

Create SLC 500 Ladder Logic for the Logic Command, Reference, and DL to Net Datalinks

1. Insert another separate rung.
2. Double-click the rung to display the rung editor.
3. Enter **MSG WRITE 500CPU LOCAL Nxx:n**, where:

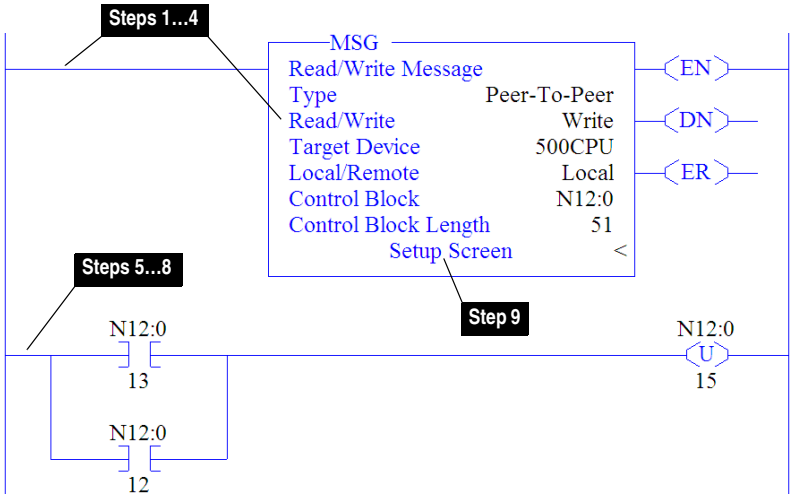
   xx is an unused data file number (for example, N12:n), and
   n is an unused element of the data file chosen for xx (for example, N12:0)
4. Press Enter.
5. Insert another separate rung.
6. Double-click the rung to display the rung editor.
7. Enter **BST XIC Nxx:n/13 NXB XIC Nxx:n/12 BND OTU Nxx:n/15**, where:

   xx and n must correspond to the assigned data file number and element (for example, N12:0) for the message created in steps 1…4.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

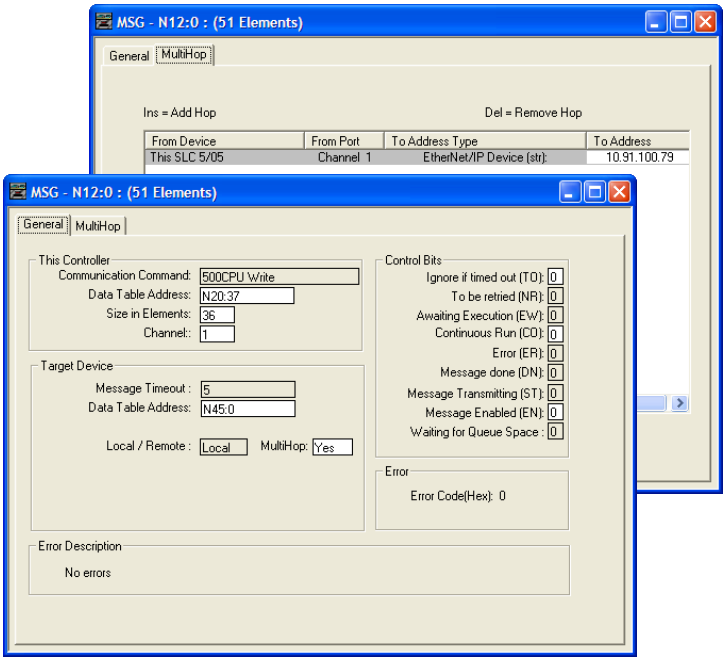
8. Press Enter.
9. In the MSG instruction (Figure 13), double-click Setup Screen to launch the message configuration dialog box (Figure 14).

Figure 13 - SLC 500 Ladder Logic for the Logic Command, Reference, and DL to Net Datalinks



10. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 14 - SLC 500 Message Configuration Dialog Boxes for the Logic Command, Reference, and DL to Net Datalinks



General Tab: This Controller	Setting
Communication Command	This setting is dimmed (unavailable) and is established when the message is created in the ladder rung.
Data Table Address (1)	**N20:37**. An unused controller data table address containing the data to be written to the drive.
Size in Elements (2)	**36**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
Channel	**1**. Controller port to which the network is connected.
General Tab: Target Device	**Setting**
Message Timeout	This setting is dimmed (unavailable). Message timeout duration in seconds.
Data Table Address (3)	**N45:0**. Specific starting address of the source file in the drive.
MultiHop	**Yes**. Enables communication to allow network messaging to be routed to the option module/drive. When 'Yes' is selected, a MultiHop tab appears on the message configuration dialog box.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.
(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.
(3) For N-File details, see N-Files on page 89.

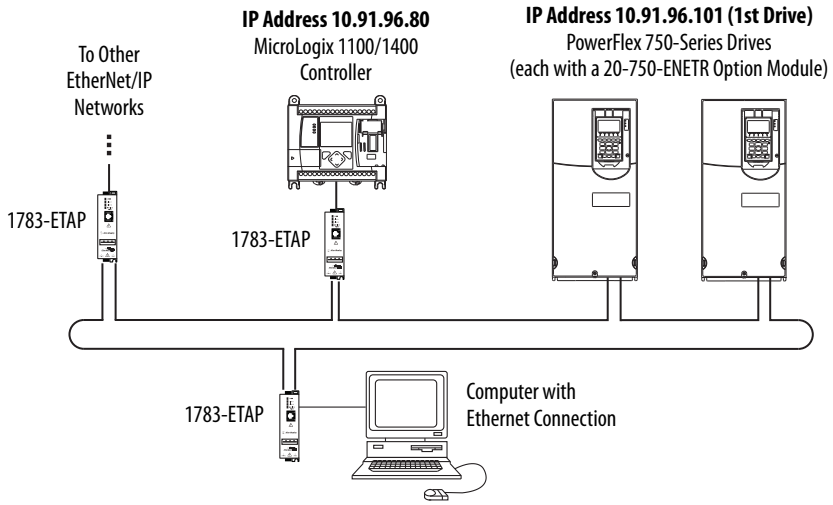
TIP This message will error out if the Control Timeout value is not changed from a non-zero value. See page 22 for writing a value to the Control Timeout.

TIP If the controller is controlling more than one drive, it is recommended to intersperse the control I/O messaging for each drive to conserve network bandwidth and decrease response time. That is, sequence the message instructions for each drive so that its group of messages will occur at a different time than those for another drive.

MicroLogix 1100/1400 Controller Example

After the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive is configured, the connected drive and option module or adapter will be a single node on the network. This section provides the steps needed to configure a simple DLR topology EtherNet/IP network (see Figure 15). In our example, we will configure a MicroLogix 1100 controller to communicate with the first drive in the network ring using Logic Command/Status, Reference/Feedback, and 32 Datalinks (16 to read data and 16 to write data) over the network.

Figure 15 - Example MicroLogix 1100/1400 Controller EtherNet/IP Device-level Ring Network



Configuring Parameters for Network I/O

Because the I/O for the drive is message-based, there is no need to configure any I/O inside the RSLogix 500 project, version 7.00 or later, until using the I/O as described in Chapter 2.

However, to get the option module or embedded adapter to operate with the I/O created in Chapter 2, we need to configure the option module to accept the I/O and drive to point to the appropriate Datalinks.

Because the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive always uses the 32-bit Logic Status and 32-bit Feedback, at least two 32-bit words must be accounted for in the controller input image. The option module or embedded adapter also uses the 32-bit Logic Command and 32-bit Reference, requiring at least two 32-bit words that must be accounted for in the controller output image. If any or all of the drive's sixteen 32-bit Datalinks are used the following applies:

- An additional 32-bit word for 'each' [**DL From Net xx**] Datalink used to write to drive or peripheral parameters must be accounted for in the controller input image.
- An additional 32-bit word for 'each' [**DL To Net xx**] Datalink used to read data must be accounted for in the controller output image.

Option Module or Embedded Adapter Parameter Settings for MicroLogix 1100 Controller Example

These option module or embedded adapter settings were used for the example ladder logic program in this section.

Option Module *Host* Parameter or Embedded Adapter Parameter	Value	Description
01 - [DL From Net 01]	370	Points to drive Par. 370 - [Stop Mode A]
02 - [DL From Net 02]	371	Points to drive Par. 371 - [Stop Mode B]
03 - [DL From Net 03]	535	Points to drive Par. 535 - [Accel Time 1]
04 - [DL From Net 04]	536	Points to drive Par. 536 - [Accel Time 2]
05 - [DL From Net 05]	537	Points to drive Par. 537 - [Decel Time 1]
06 - [DL From Net 06]	538	Points to drive Par. 538 - [Decel Time 2]
07 - [DL From Net 07]	539	Points to drive Par. 539 - [Jog Acc Dec Time]
08 - [DL From Net 08]	556	Points to drive Par. 556 - [Jog Speed 1]
09 - [DL From Net 09]	557	Points to drive Par. 557 - [Jog Speed 2]
10 - [DL From Net 10]	571	Points to drive Par. 571 - [Preset Speed 1]
11 - [DL From Net 11]	572	Points to drive Par. 572 - [Preset Speed 2]
12 - [DL From Net 12]	573	Points to drive Par. 573 - [Preset Speed 3]
13 - [DL From Net 13]	574	Points to drive Par. 574 - [Preset Speed 4]
14 - [DL From Net 14]	575	Points to drive Par. 575 - [Preset Speed 5]
15 - [DL From Net 15]	576	Points to drive Par. 576 - [Preset Speed 6]
16 - [DL From Net 16]	577	Points to drive Par. 577 - [Preset Speed 7]
17 - [DL To Net 01]	370	Points to drive Par. 370 - [Stop Mode A]
18 - [DL To Net 02]	371	Points to drive Par. 371 - [Stop Mode B]
19 - [DL To Net 03]	535	Points to drive Par. 535 - [Accel Time 1]
20 - [DL To Net 04]	536	Points to drive Par. 536 - [Accel Time 2]
21 - [DL To Net 05]	537	Points to drive Par. 537 - [Decel Time 1]
22 - [DL To Net 06]	538	Points to drive Par. 538 - [Decel Time 2]
23 - [DL To Net 07]	539	Points to drive Par. 539 - [Jog Acc Dec Time]
24 - [DL To Net 08]	556	Points to drive Par. 556 - [Jog Speed 1]
25 - [DL To Net 09]	557	Points to drive Par. 557 - [Jog Speed 2]
26 - [DL To Net 10]	571	Points to drive Par. 571 - [Preset Speed 1]
27 - [DL To Net 11]	572	Points to drive Par. 572 - [Preset Speed 2]
28 - [DL To Net 12]	573	Points to drive Par. 573 - [Preset Speed 3]
29 - [DL To Net 13]	574	Points to drive Par. 574 - [Preset Speed 4]
30 - [DL To Net 14]	575	Points to drive Par. 575 - [Preset Speed 5]
31 - [DL To Net 15]	576	Points to drive Par. 576 - [Preset Speed 6]
32 - [DL To Net 16]	577	Points to drive Par. 577 - [Preset Speed 7]

TIP The **[DL From Net xx]** parameters are inputs into the drive that come from controller outputs (for example, data to write to a drive parameter). The **[DL To Net xx]** parameters are outputs from the drive that go to controller inputs (for example, data to read a drive parameter).

Creating RSLogix 500 Project, Version 7 or Later

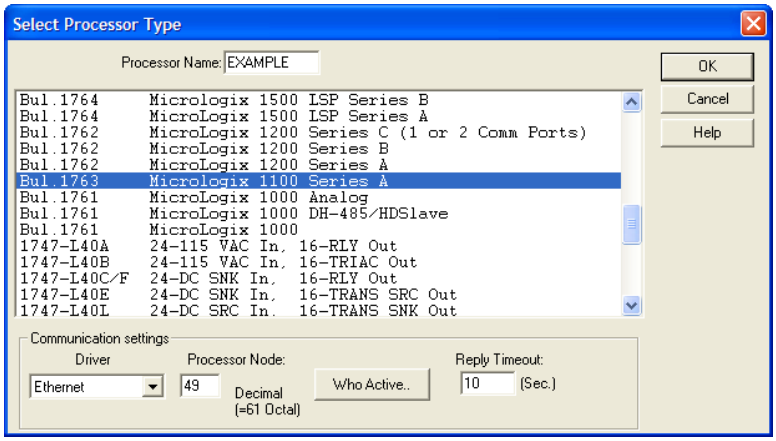
To transmit (read and write) data between the controller and drive, you must create message instructions that allocate data table addresses in the controller for Logic Command/Status, Reference/Feedback, and Datalinks. Note that three messages need to be configured. The timeout message has to be executed first before the Logic Command, Reference, and DL To/From Net Datalink messages will work. For more information on N42:3 and N45 target device data table addresses, see N-Files on page 89.

Select the Controller

1. Start RSLogix 500 software.

   The RSLogix 500 dialog box appears.

2. From the File menu, choose New to display the Select Processor Type dialog box.



3. Assign a name for the processor.
4. In the list, select the MicroLogix 1100.
5. Choose the appropriate choices for the fields in the dialog box to match your application.
6. Click OK.

   The RSLogix 500 project dialog box appears.

Create MicroLogix 1100/1400 Ladder Logic for the Control Timeout

1. In the RSLogix 500 project dialog box treeview under Program Files double-click LAD 2.
2. Insert a ladder rung.
3. Double-click the rung to display the rung editor.
4. Enter **MSG MGxx:n**, where:

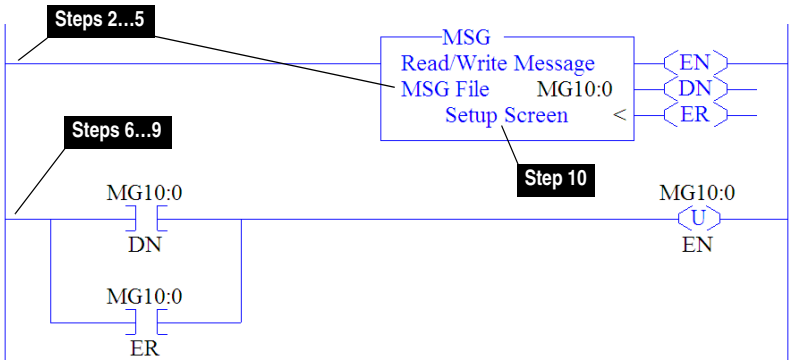
   xx is an unused data file number (for example, MG10:n), and
   n is an unused element of the data file chosen for xx (for example, MG10:0)
5. Press Enter.
6. Insert another separate rung.
7. Double-click the rung to display the rung editor.
8. Enter **BST XIC MGxx:n/DN NXB XIC MGxx:n/ER BND OTU MGxx:n/EN**, where:

   xx and n must correspond to the assigned data file number and element (for example, MG10:0) for the message created in steps 2…5.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

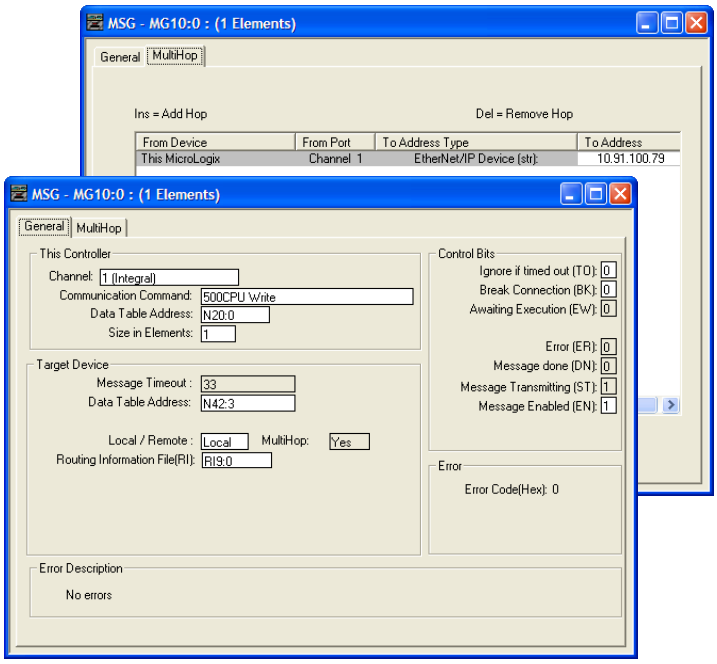
9. Press Enter.
10. In the MSG instruction (Figure 16), double-click Setup Screen to launch the message configuration dialog box (Figure 17).

Figure 16 - MicroLogix 1100/1400 Ladder Logic for the Control Timeout



11. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 17 - MicroLogix 1100/1400 Message Configuration Dialog Boxes for the Control Timeout



General Tab: This Controller	Setting
Channel	**1 (integral)**. Controller port to which the network is connected.
Communication Command	**500CPU Write**. The controller type and command type for the controller to read or write data. Because the MicroLogix 1100 is part of the SLC 500 controller family, the '500CPU' controller type was selected. The 'Write' command type was selected to write the control timeout value to the drive.
Data Table Address (1)	**N20:0**. An unused controller data table address containing the control timeout value to be written.
Size in Elements (2)	**1**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
General Tab: Target Device	**Setting**
Message Timeout	**5**. Message timeout duration in seconds.
Data Table Address (3)	**N42:3**. Specific starting address of the destination file in the drive.
Routing Information File	**RI9:0**. An unused routing information file for the controller.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.

(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.

(3) For details on setting the control timeout value and its function, see N-Files on page 89. The Control Timeout (N42:3) is stored in RAM. If the EtherNet/IP option module is power cycled, the Control Timeout Message must be re-sent. If the Control Timeout is not changed from a non-zero value, the control message (page 35) will error out.

TIP The Control Timeout (N42:3) must be changed to a non-zero value (5…20 seconds recommended). If the Control Timeout is not changed from a non-zero value, the control message (page 35) will error out. The Control Timeout is stored in RAM. If the option module is power cycled, the Control Timeout Message must be re-sent.

Create MicroLogix 1100/1400 Ladder Logic for the Logic Status, Feedback, and DL from Net Datalinks

1. Insert another separate rung.
2. Double-click the rung to display the rung editor.
3. Enter **MSG MGxx:n**, where:

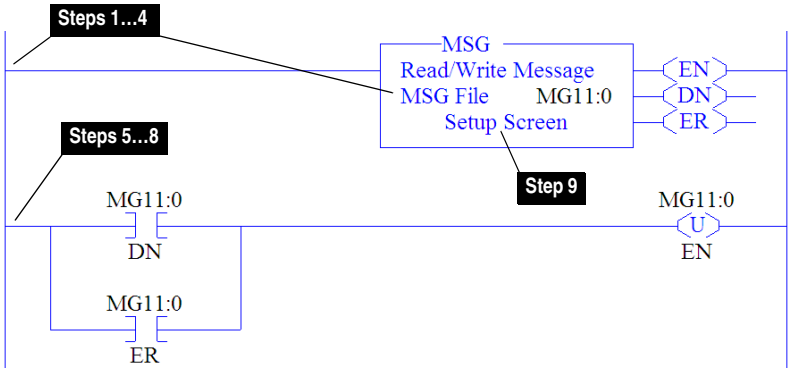
   xx is an unused data file number (for example, MG11:n), and
   n is an unused element of the data file chosen for xx (for example, MG11:0)
4. Press Enter.
5. Insert another separate rung.
6. Double-click the rung to display the rung editor.
7. Enter **BST XIC MGxx:n/DN NXB XIC MGxx:n/ER BND OTU MGxx:n/EN**, where:

   xx and n must correspond to the assigned data file number and element (for example, MG11:0) for the message created in steps 1…4.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

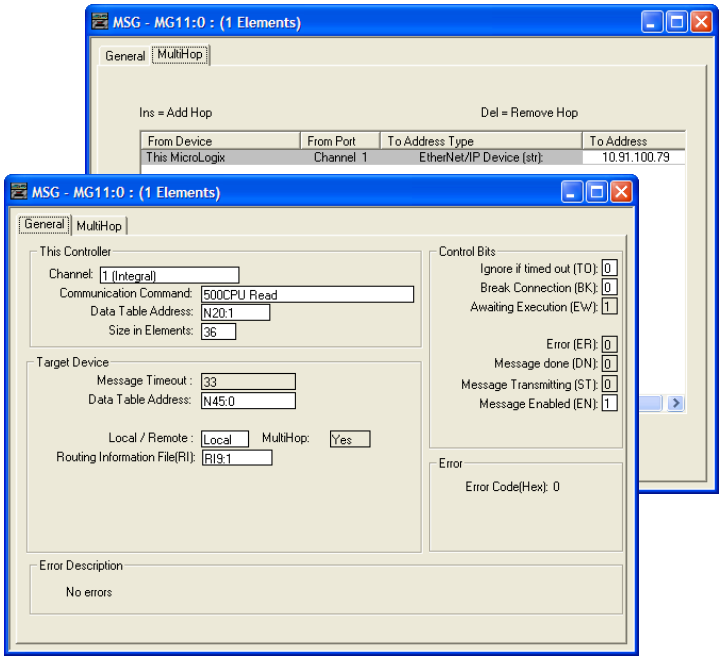
8. Press Enter.
9. In the MSG instruction (Figure 18), double-click Setup Screen to launch the message configuration dialog box (Figure 19).

Figure 18 - MicroLogix 1100/1400 Ladder Logic for the Logic Status, Feedback, and DL from Net Datalinks



10. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 19 - MicroLogix 1100/1400 Message Configuration Dialog Boxes for the Logic Status, Feedback, and DL from Net Datalinks



General Tab: This Controller	Setting
Channel	**1 (integral)**. Controller port to which the network is connected.
Communication Command	**500CPU Read**. The controller type and command type for the controller to read or write data. Because the MicroLogix 1100 is part of the SLC 500 controller family, the '500CPU' controller type was selected. The 'Read' command type was selected to read data from the drive.
Data Table Address (1)	**N20:1**. An unused controller data table address containing the data to be read from the drive.
Size in Elements (2)	**36**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
General Tab: Target Device	**Setting**
Message Timeout	**5**. Message timeout duration in seconds.
Data Table Address (3)	**N45:0**. Specific starting address of the source file in the drive.
Routing Information File	**RI9:1**. An unused routing information file for the controller.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.
(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.
(3) For N-File details, see N-Files on page 89.

Create MicroLogix 1100/1400 Ladder Logic for the Logic Command, Reference, and DL to Net Datalinks

1. Insert another separate rung.
2. Double-click the rung to display the rung editor.
3. Enter **MSG MGxx:n**, where:

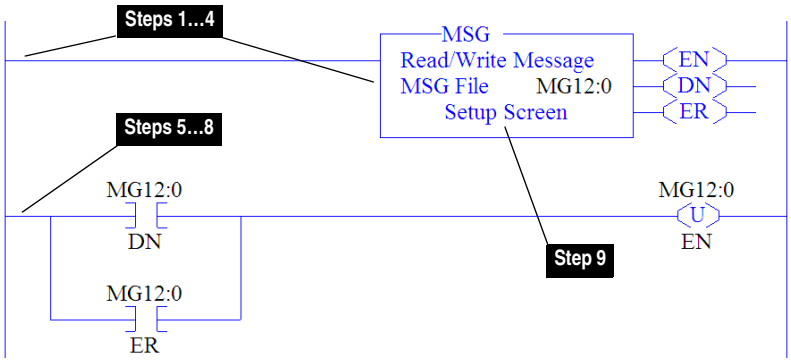
   xx is an unused data file number (for example, MG12:n), and
   n is an unused element of the data file chosen for xx (for example, MG12:0)
4. Press Enter.
5. Insert another separate rung.
6. Double-click the rung to display the rung editor.
7. Enter **BST XIC MGxx:n/DN NXB XIC MGxx:n/ER BND OTU MGxx:n/EN**, where:

   xx and n must correspond to the assigned data file number and element (for example, MG12:0) for the message created in steps 1…4.

IMPORTANT The information must be entered with appropriate numbers for 'xx' and 'n' for your application, and with spaces and forward slashes exactly as shown.

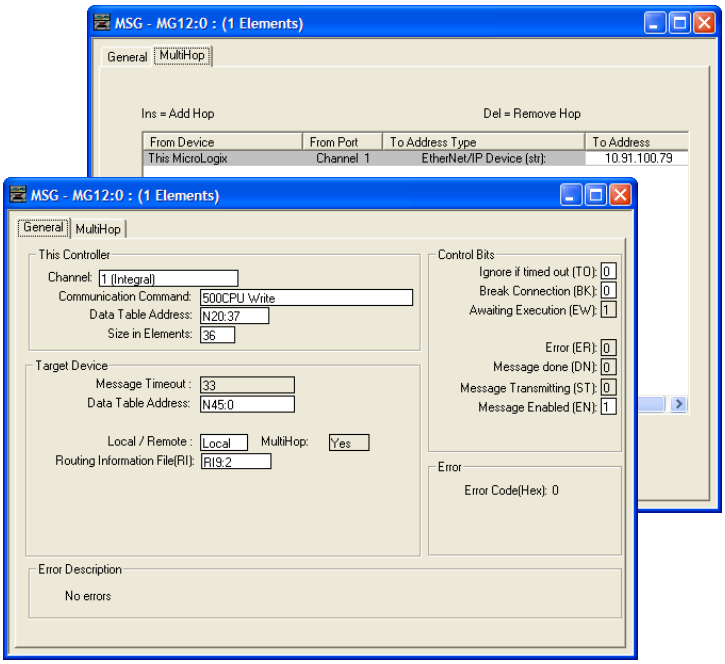
8. Press Enter.
9. In the MSG instruction (Figure 20), double-click Setup Screen to launch the message configuration dialog box (Figure 21).

Figure 20 - MicroLogix 1100/1400 Ladder Logic for the Logic Command, Reference, and DL to Net Datalinks



10. Configure the General tab fields by entering or verifying the information shown in the message configuration dialog box.

Figure 21 - MicroLogix 1100/1400 Message Configuration Dialog Boxes for the Logic Command, Reference, and DL to Net Datalinks



General Tab: This Controller	Setting
Channel	**1 (integral)**. Controller port to which the network is connected.
Communication Command	**500CPU Write**. The controller type and command type for the controller to read or write data. Because the MicroLogix 1100 is part of the SLC 500 controller family, the '500CPU' controller type was selected. The 'Write' command type was selected to write data to the drive.
Data Table Address (1)	**N20:37**. An unused controller data table address containing the data to be written to the drive.
Size in Elements (2)	**36**. Number of elements (words) to be transferred. Each element size is a 16-bit integer.
General Tab: Target Device	**Setting**
Message Timeout	**5**. Message timeout duration in seconds.
Data Table Address (3)	**N45:0**. Specific starting address of the destination file in the drive.
Routing Information File	**RI9:2**. An unused routing information file for the controller.
MultiHop Tab	**Setting**
To Address	**10.91.100.79.** The IP address of the option module connected to the drive.

(1) For details on data table addresses for this example project, see Table 3 on page 43.
(2) For details to determine element size for a specific drive, see Understanding Controller Data Table Addresses on page 42.
(3) For N-File details, see N-Files on page 89.

TIP This message will error out if the Control Timeout value is not changed from a non-zero value. See page 31 for writing a value to the Control Timeout.

TIP If the controller is controlling more than one drive, it is recommended to intersperse the control I/O messaging for each drive to conserve network bandwidth and decrease response time. That is, sequence the message instructions for each drive so that its group of messages will occur at a different time than those for another drive.

Using the I/O

This chapter provides information and examples that explain how to control, configure, and monitor a PowerFlex 750-Series drive using the configured I/O.

ATTENTION: Risk of injury or equipment damage exists. The examples in this publication are intended solely for purposes of example. There are many variables and requirements with any application. Rockwell Automation, Inc. does not assume responsibility or liability (to include intellectual property liability) for actual use of the examples shown in this publication.

About I/O Messaging

On CIP-based networks, including EtherNet/IP, I/O connections are used to transfer the data that controls the PowerFlex drive and sets its Reference. I/O can also be used to transfer data to and from Datalinks in PowerFlex 750-Series drives.

The network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive includes the Logic Command, Logic Status, Reference, Feedback, and memory allocation for the Generic Ethernet module profile (all as 32-bit words) in the controller's I/O image. This basic I/O must always be configured in the EtherNet bridge using RSLogix software. Additional I/O, if needed, can be set using up to 16 Datalinks to write data and/or up to 16 Datalinks to read data. When using any combination of these Datalinks, add one 32-bit word for **each** Datalink to the basic I/O Input Size and/or Output Size.

Chapter 1, Configuring the I/O, discusses how to configure the option module or embedded adapter and controller on the network for the required I/O. The Glossary defines the different options. This chapter discusses how to use I/O after you have configured the option module or embedded adapter and controller.

Understanding the I/O Image

The terms 'input' and 'output' are defined from the controller's point of view. Therefore, output I/O is data that is produced by the controller and consumed by the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive. Input I/O is status data that is produced by the option module or embedded adapter and consumed as input by the controller. The I/O image will vary based on how many of the drive's 32-bit Datalinks (**DL From Net 01-16** and **DL To Net 01-16**) are used.

The I/O image for the PLC-5, SLC 500, and MicroLogix 1100/1400 controller changes depending on how many of the drive's sixteen 32-bit Datalinks are used.

TIP Because PLC-5, SLC 500, and MicroLogix 1100/1400 controllers are 16-bit devices, each 32-bit word for the Logic Command/Status, Reference/Feedback, and any used Datalinks will consume two contiguous words (Least Significant Word and Most Significant Word) in the controller's I/O image. Table 1 shows the I/O for a drive using all 32-bit Datalinks.

Table 1 - PLC-5, SLC 500, and MicroLogix 1100/1400 Controller I/O Image for PowerFlex 750-Series Drives (32-bit Logic Command/Status, Reference/Feedback, and Datalinks)

Word	Output I/O
0	Logic Command (LSW)
1	Logic Command (MSW)
2	Reference (LSW)
3	Reference (MSW)
4	DL From Net 01 (LSW)
5	DL From Net 01 (MSW)
6	DL From Net 02 (LSW)
7	DL From Net 02 (MSW)
8	DL From Net 03 (LSW)
9	DL From Net 03 (MSW)
10	DL From Net 04 (LSW)
11	DL From Net 04 (MSW)
12	DL From Net 05 (LSW)
13	DL From Net 05 (MSW)
14	DL From Net 06 (LSW)
15	DL From Net 06 (MSW)
16	DL From Net 07 (LSW)
17	DL From Net 07 (MSW)
18	DL From Net 08 (LSW)
19	DL From Net 08 (MSW)
20	DL From Net 09 (LSW)
21	DL From Net 09 (MSW)
22	DL From Net 10 (LSW)
23	DL From Net 10 (MSW)
24	DL From Net 11 (LSW)
25	DL From Net 11 (MSW)
26	DL From Net 12 (LSW)
27	DL From Net 12 (MSW)
28	DL From Net 13 (LSW)
29	DL From Net 13 (MSW)
30	DL From Net 14 (LSW)

Word	Input I/O
0	Logic Status (LSW)
1	Logic Status (MSW)
2	Feedback (LSW)
3	Feedback (MSW)
4	DL To Net 01 (LSW)
5	DL To Net 01 (MSW)
6	DL To Net 02 (LSW)
7	DL To Net 02 (MSW)
8	DL To Net 03 (LSW)
9	DL To Net 03 (MSW)
10	DL To Net 04 (LSW)
11	DL To Net 04 (MSW)
12	DL To Net 05 (LSW)
13	DL To Net 05 (MSW)
14	DL To Net 06 (LSW)
15	DL To Net 06 (MSW)
16	DL To Net 07 (LSW)
17	DL To Net 07 (MSW)
18	DL To Net 08 (LSW)
19	DL To Net 08 (MSW)
20	DL To Net 09 (LSW)
21	DL To Net 09 (MSW)
22	DL To Net 10 (LSW)
23	DL To Net 10 (MSW)
24	DL To Net 11 (LSW)
25	DL To Net 11 (MSW)
26	DL To Net 12 (LSW)
27	DL To Net 12 (MSW)
28	DL To Net 13 (LSW)
29	DL To Net 13 (MSW)
30	DL To Net 14 (LSW)

Word	Output I/O
31	DL From Net 14 (MSW)
32	DL From Net 15 (LSW)
33	DL From Net 15 (MSW)
34	DL From Net 16 (LSW)
35	DL From Net 16 (MSW)

Word	Input I/O
31	DL To Net 14 (MSW)
32	DL To Net 15 (LSW)
33	DL To Net 15 (MSW)
34	DL To Net 16 (LSW)
35	DL To Net 16 (MSW)

Using Logic Command/Status

The 'Logic Command' is a 32-bit word of control data produced by the controller and consumed by the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive. The 'Logic Status' is a 32-bit word of status data produced by the option module and consumed by the controller.

For a PLC-5, SLC 500 or MicroLogix 1100/1400 controller, the Logic Command word is always words 0 (Least Significant Word) and 1 (Most Significant Word) in the output image and the Logic Status word is always words 0 (least Significant Word) and 1 (Most Significant Word) in the input image.

This manual contains the bit definitions for compatible products available at the time of publication in Appendix B, Logic Command/Status Words: PowerFlex 750-Series Drives.

Using Reference/Feedback

The 'Reference' is a 32-bit REAL (floating point) piece of control data produced by the controller and consumed by the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive. The 'Feedback' is a 32-bit REAL (floating point) piece of status data produced by the option module or embedded adapter and consumed by the controller.

For a PLC-5, SLC 500 or MicroLogix 1100/1400 controller, the 32-bit REAL Reference word is always words 2 (Least Significant Word) and 3 (Most Significant Word) in the output image and the 32-bit REAL Feedback is always words 2 (Least Significant Word) and 3 (Most Significant Word) in the input image.

The Reference and Feedback 32-bit REAL value represents drive speed. The scaling for the speed Reference and Feedback are dependent on drive Parameter 300 - [Speed Units]. For example, if Parameter 300 is set to Hz, a 32-bit REAL Reference value of '30.0' would equal a Reference of 30.0 Hz. If Parameter 300 is set to RPM, a 32-bit REAL Reference value of '1020.5' would equal a Reference of 1020.5 RPM. Note that the commanded maximum speed can never exceed the value of drive Parameter 520 - [Max Fwd Speed]. Table 2 shows example References and their results for a PowerFlex 755 drive that has the following:

- Parameter 300 - [Speed Units] set to Hz.
- Parameter 37 - [Maximum Freq] set to 130 Hz.
- Parameter 520 - [Max Fwd Speed] set to 60 Hz.

When Parameter 300 - [Speed Units] is set to RPM, the other parameters are also in RPM.

Table 2 - PowerFlex 750-Series Drive Example Speed Reference/Feedback Scaling

Network Reference Value	Speed Command Value (2)	Output Speed	Network Feedback Value
130.0	130 Hz	60 Hz (3)	60.0
65.0	65 Hz	60 Hz (3)	60.0
32.5	32.5 Hz	32.5 Hz	32.5
0.0	0 Hz	0 Hz	0.0
-32.5 (1)	32.5 Hz	32.5 Hz	32.5

(1) The effects of values less than 0.0 depend on whether the PowerFlex 750-Series drive uses a bipolar or unipolar direction mode. See the drive documentation for details.
(2) For this example, drive Parameter 300 - [Speed Units] is set to Hz.
(3) The drive runs at 60 Hz instead of 130 Hz or 65 Hz because drive Parameter 520 - [Max Fwd Speed] sets 60 Hz as the maximum speed.

Using Datalinks

A Datalink is a mechanism used by PowerFlex drives to transfer data to and from the controller. Datalinks allow a drive parameter value to be read or written to without using an Explicit Message. When enabled, each Datalink occupies two 16-bit words in a PLC-5, SLC 500 or MicroLogix 1100/1400 controller.

The following rules apply when using PowerFlex 750-Series drive Datalinks:

- The target of a Datalink can be any 'Host' parameter, including those of a peripheral. For example, drive parameter 535 - [Accel Time 1] can be the target of any or all option modules installed in the drive.
- The data passed through the drive's Datalink mechanism is determined by the settings of 'Host' **Parameters 01…16 - [DL From Net 01-16]** and 'Host' **Parameters 17…32 - [DL To Net 01-16]**.

  IMPORTANT A reset is always required after configuring Datalinks so that the changes take effect.

- When an I/O connection that includes Datalinks is active, those Datalinks being used are locked and cannot be changed until that I/O connection becomes idle or inactive.
- When you use a Datalink to change a value, the value is **not** written to the Nonvolatile Storage (NVS). The value is stored in volatile memory and lost when the drive loses power. Thus, use Datalinks when you need to change a value of a parameter frequently.

Datalinks for PowerFlex 750-Series drive peripherals (embedded EtherNet/IP adapter on PowerFlex 755 drives only and option modules such as an encoder or a communication module) are locked when the peripheral has an I/O connection with a controller. When a controller has an I/O connection to the drive, the drive does not allow a reset to defaults, configuration download or anything else that

could change the makeup of the I/O connection in a running system. The I/O connection with the controller must first be disabled to allow changes to the respective Datalinks.

Depending on the controller being used, the I/O connection can be disabled by doing the following:

- Inhibiting the module in RSLogix software
- Putting the controller in Program mode
- Placing the scanner in idle mode
- Disconnecting the drive from the network

DeviceLogix Datalinks are also locked while the DeviceLogix program is running. The DeviceLogix program must first be disabled to allow changes to the Datalinks. Set DeviceLogix parameter 53 - [DLX Operation] to 'DisableLogic' to disable the logic (the parameter value will then change to 'LogicDisabld').

TIP A COP (Copy) instruction or a UDDT is needed—for REAL parameters, speed Reference, and speed Feedback only—to copy the DINT data into a REAL word for input data conversion. For output data conversion, a COP (Copy) instruction or UDDT is needed—for REAL parameters, speed Reference, and speed Feedback only—to copy the REAL data into a DINT word. To determine whether a parameter is a 32-bit integer (DINT) or a REAL data type, see the Data Type column in the chapter containing parameters in the PowerFlex 750-Series AC Drives Programming Manual (publication 750-PM001).

Example Ladder Logic Program Information

The example ladder logic programs in this chapter are intended for and operate PowerFlex 750-Series drives.

Functions of the Example Programs

The example programs enable you to do the following:

- Receive Logic Status information from the drive.
- Send a Logic Command to control the drive (for example, start, stop).
- Send a Reference to the drive and receive Feedback from the drive.
- Send/receive Datalink data to/from the drive.

Logic Command/Status Words

These examples use the Logic Command word and Logic Status word for PowerFlex 750-Series drives. See Appendix B, Logic Command/Status Words: PowerFlex 750-Series Drives, to view details.

PLC-5, SLC 500, and MicroLogix 1100/1400 Controller Example

The information in this section is common for and applies to PLC-5, SLC 500, and MicroLogix 1100/1400 controllers.

Option Module or Embedded Adapter Parameter Settings

For option module or embedded adapter settings used for the example ladder logic program in this section, see the following pages.

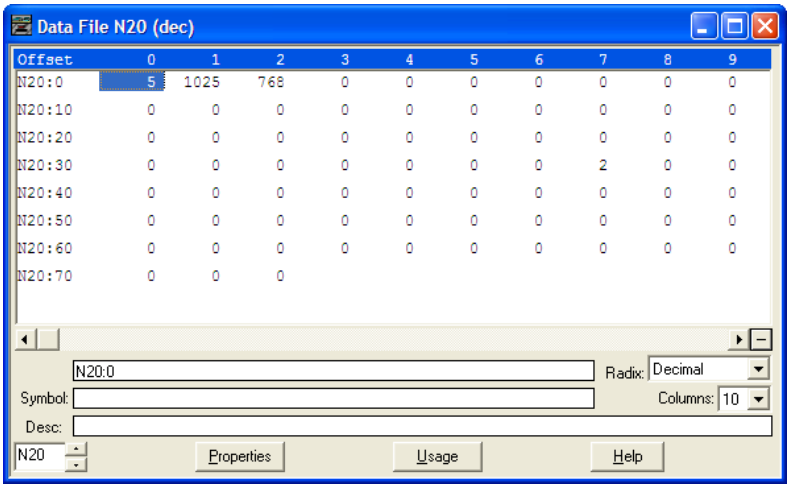
Controller	See Table on
PLC-5	page 11
SLC 500	page 20
MicroLogix 1100/1400	page 29

Understanding Controller Data Table Addresses

Because PLC-5, SLC 500, and MicroLogix 1100/1400 controllers are 16-bit platforms being used with the 32-bit EtherNet/IP option module, the data will be transposed from the least significant word (LSW) to the most significant word (MSW) in the controller.

When the I/O was configured (Chapter 1), an available data table file (N20) was used. Figure 22 shows the entire data file address structure for this example.

Figure 22 - Data File Table for Example Ladder Logic Program



IMPORTANT The N20:0 data table address in this example is used to set a control timeout value (in seconds), which determines how long it will take the option module to detect a communication loss. Enter a valid value (1...32767) for N20:0. A value of zero (0) is not valid, because it disables the timeout and all I/O messages (Logic Command/Status, Reference/Feedback, and Datalinks) intended for the drive will not execute. A timeout value of 5...20 seconds is recommended.

Table 3 shows the I/O definitions as they relate to the N20 data table file (Figure 22) being used in this example.

For PowerFlex 750-Series drives, which contain both DINT (32-bit format) and REAL (floating point format) data types, you will always first read from and write to the LSW data table address in the controller. Then if the data value exceeds 16 bits, the remaining value will be in the MSW data table address.

Table 3 - PLC-5, SLC 500, and MicroLogix 1100/1400 Controller Data Table Addresses for PowerFlex 750-Series Drives

Data Table Address	Description
N20:0	Control Timeout
N20:1	Logic Status (LSW, see Appendix B)
N20:2	Logic Status (MSW, see Appendix B)
N20:3	Speed Feedback LSW
N20:4	Speed Feedback MSW
N20:5	Value of parameter assigned to option module *Host* Parameter 17 [DL To Net 01] LSW
N20:6	Value of parameter assigned to option module *Host* Parameter 17 [DL To Net 01] MSW
N20:7	Value of parameter assigned to option module *Host* Parameter 18 [DL To Net 02] LSW
N20:8	Value of parameter assigned to option module *Host* Parameter 18 [DL To Net 02] MSW
N20:9	Value of parameter assigned to option module *Host* Parameter 19 [DL To Net 03] LSW
N20:10	Value of parameter assigned to option module *Host* Parameter 19 [DL To Net 03] MSW
N20:11	Value of parameter assigned to option module *Host* Parameter 20 [DL To Net 04] LSW
N20:12	Value of parameter assigned to option module *Host* Parameter 20 [DL To Net 04] MSW
N20:13	Value of parameter assigned to option module *Host* Parameter 21 [DL To Net 05] LSW
N20:14	Value of parameter assigned to option module *Host* Parameter 21 [DL To Net 05] MSW
N20:15	Value of parameter assigned to option module *Host* Parameter 22 [DL To Net 06] LSW
N20:16	Value of parameter assigned to option module *Host* Parameter 22 [DL To Net 06] MSW
N20:17	Value of parameter assigned to option module *Host* Parameter 23 [DL To Net 07] LSW
N20:18	Value of parameter assigned to option module *Host* Parameter 23 [DL To Net 07] MSW
N20:19	Value of parameter assigned to option module *Host* Parameter 24 [DL To Net 08] LSW
N20:20	Value of parameter assigned to option module *Host* Parameter 24 [DL To Net 08] MSW
N20:21	Value of parameter assigned to option module *Host* Parameter 25 [DL To Net 09] LSW
N20:22	Value of parameter assigned to option module *Host* Parameter 25 [DL To Net 09] MSW
N20:23	Value of parameter assigned to option module *Host* Parameter 26 [DL To Net 10] LSW
N20:24	Value of parameter assigned to option module *Host* Parameter 26 [DL To Net 10] MSW
N20:25	Value of parameter assigned to option module *Host* Parameter 27 [DL To Net 11] LSW
N20:26	Value of parameter assigned to option module *Host* Parameter 27 [DL To Net 11] MSW
N20:27	Value of parameter assigned to option module *Host* Parameter 28 [DL To Net 12] LSW
N20:28	Value of parameter assigned to option module *Host* Parameter 28 [DL To Net 12] MSW
N20:29	Value of parameter assigned to option module *Host* Parameter 29 [DL To Net 13] LSW
N20:30	Value of parameter assigned to option module *Host* Parameter 29 [DL To Net 13] MSW
N20:31	Value of parameter assigned to option module *Host* Parameter 30 [DL To Net 14] LSW
N20:32	Value of parameter assigned to option module *Host* Parameter 30 [DL To Net 14] MSW
N20:33	Value of parameter assigned to option module *Host* Parameter 31 [DL To Net 15] LSW
N20:34	Value of parameter assigned to option module *Host* Parameter 31 [DL To Net 15] MSW
N20:35	Value of parameter assigned to option module *Host* Parameter 32 [DL To Net 16] LSW
N20:36	Value of parameter assigned to option module *Host* Parameter 32 [DL To Net 16] MSW
N20:37	Logic Command (LSW, see Appendix B)
N20:38	Logic Command (MSW, see Appendix B)
N20:39	Speed Reference LSW

Table 3 - PLC-5, SLC 500, and MicroLogix 1100/1400 Controller Data Table Addresses for PowerFlex 750-Series Drives (Continued)

Data Table Address	Description
N20:40	Speed Reference MSW
N20:41	Value of parameter assigned to option module *Host* Parameter 01 [DL From Net 01] LSW
N20:42	Value of parameter assigned to option module *Host* Parameter 01 [DL From Net 01] MSW
N20:43	Value of parameter assigned to option module *Host* Parameter 02 [DL From Net 02] LSW
N20:44	Value of parameter assigned to option module *Host* Parameter 02 [DL From Net 02] MSW
N20:45	Value of parameter assigned to option module *Host* Parameter 03 [DL From Net 03] LSW
N20:46	Value of parameter assigned to option module *Host* Parameter 03 [DL From Net 03] MSW
N20:47	Value of parameter assigned to option module *Host* Parameter 04 [DL From Net 04] LSW
N20:48	Value of parameter assigned to option module *Host* Parameter 04 [DL From Net 04] MSW
N20:49	Value of parameter assigned to option module *Host* Parameter 05 [DL From Net 05] LSW
N20:50	Value of parameter assigned to option module *Host* Parameter 05 [DL From Net 05] MSW
N20:51	Value of parameter assigned to option module *Host* Parameter 06 [DL From Net 06] LSW
N20:52	Value of parameter assigned to option module *Host* Parameter 06 [DL From Net 06] MSW
N20:53	Value of parameter assigned to option module *Host* Parameter 07 [DL From Net 07] LSW
N20:54	Value of parameter assigned to option module *Host* Parameter 07 [DL From Net 07] MSW
N20:55	Value of parameter assigned to option module *Host* Parameter 08 [DL From Net 08] LSW
N20:56	Value of parameter assigned to option module *Host* Parameter 08 [DL From Net 08] MSW
N20:57	Value of parameter assigned to option module *Host* Parameter 09 [DL From Net 09] LSW
N20:58	Value of parameter assigned to option module *Host* Parameter 09 [DL From Net 09] MSW
N20:59	Value of parameter assigned to option module *Host* Parameter 10 [DL From Net 10] LSW
N20:60	Value of parameter assigned to option module *Host* Parameter 10 [DL From Net 10] MSW
N20:61	Value of parameter assigned to option module *Host* Parameter 11 [DL From Net 11] LSW
N20:62	Value of parameter assigned to option module *Host* Parameter 11 [DL From Net 11] MSW
N20:63	Value of parameter assigned to option module *Host* Parameter 12 [DL From Net 12] LSW
N20:64	Value of parameter assigned to option module *Host* Parameter 12 [DL From Net 12] MSW
N20:65	Value of parameter assigned to option module *Host* Parameter 13 [DL From Net 13] LSW
N20:66	Value of parameter assigned to option module *Host* Parameter 13 [DL From Net 13] MSW
N20:67	Value of parameter assigned to option module *Host* Parameter 14 [DL From Net 14] LSW
N20:68	Value of parameter assigned to option module *Host* Parameter 14 [DL From Net 14] MSW
N20:69	Value of parameter assigned to option module *Host* Parameter 15 [DL From Net 15] LSW
N20:70	Value of parameter assigned to option module *Host* Parameter 15 [DL From Net 15] MSW
N20:71	Value of parameter assigned to option module *Host* Parameter 16 [DL From Net 16] LSW
N20:72	Value of parameter assigned to option module *Host* Parameter 16 [DL From Net 16] MSW

TIP Remember that most of the parameters in the drive being read/written with the Datalinks are REAL (floating point) data types. Therefore, use a COP (Copy) instruction to convert the least significant word and most significant word values to a single floating point register (Fx:x).

You can use the controller data table addresses to directly control and monitor the drive without creating any ladder logic program. However, if you intend to use Human Machine Interface devices (PanelView, and so forth) to operate the drive and view its status, you may want to create alternate controller data table addresses (Table 4 and Table 5) and a ladder logic program that will pass that data to the data table addresses used for messaging.

Table 4 - Controller and Program Data Table Address Descriptions for Example Logic Status/ Feedback Ladder Logic Program

Description	Controller Data Table Address
Drive Ready	N20:1/0
Drive Active	N20:1/1
Actual Direction Forward (XIO)	N20:1/3
Actual Direction Reverse (XIC)	N20:1/3
Drive Faulted	N20:1/7
Drive At Speed	N20:1/8
Speed Feedback	N20:3

Description	Program Data Table Address
Status Ready	B3:1/0
Status Active	B3:1/1
Status Forward	B3:1/3
Status Reverse	B3:1/4
Status Faulted	B3:1/7
Status At Speed	B3:1/8
Speed Feedback	B30:3

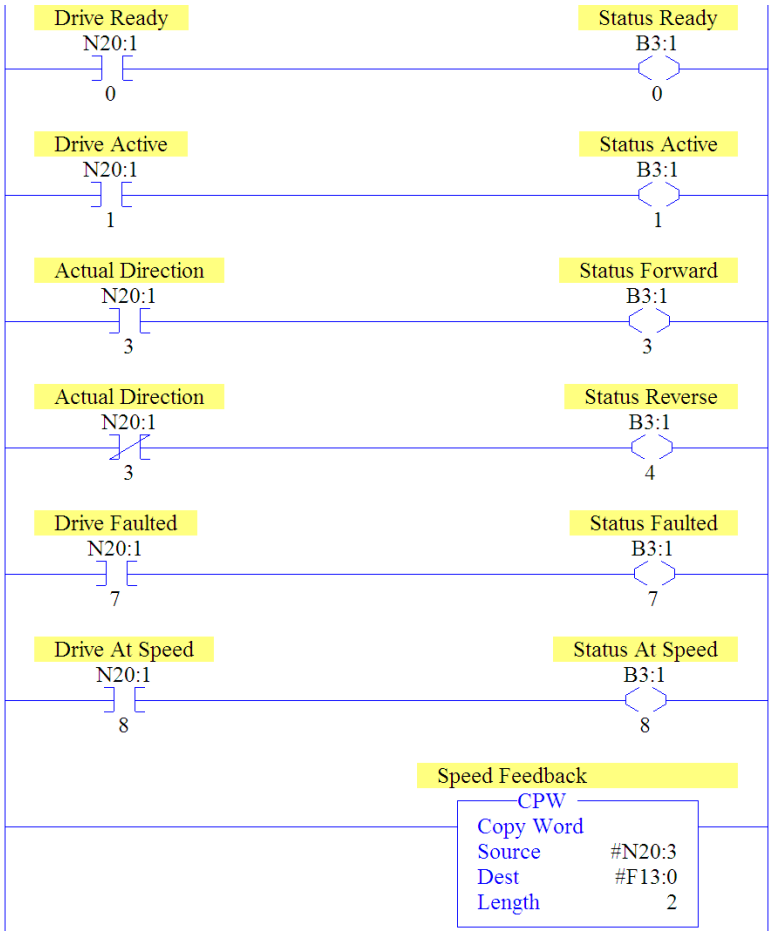
Table 5 - Program and Controller Data Table Address Descriptions for Example Logic Command/ Reference Ladder Logic Program

Description	Program Data Table Address
Command Stop	B3:20/0
Command Start	B3:20/1
Command Jog	B3:20/2
Command Clear Faults	B3:20/3
Command Forward Reverse (XIO)	B3:20/4
Command Forward Reverse (XIC)	B3:20/4
Speed Reference	N30:22

Description	Controller Data Table Address
Drive Stop	N20:20/0
Drive Start	N20:20/1
Drive Jog	N20:20/2
Drive Clear Faults	N20:20/3
Drive Forward	N20:20/4
Drive Reverse	N20:20/5
Speed Reference	N20:22

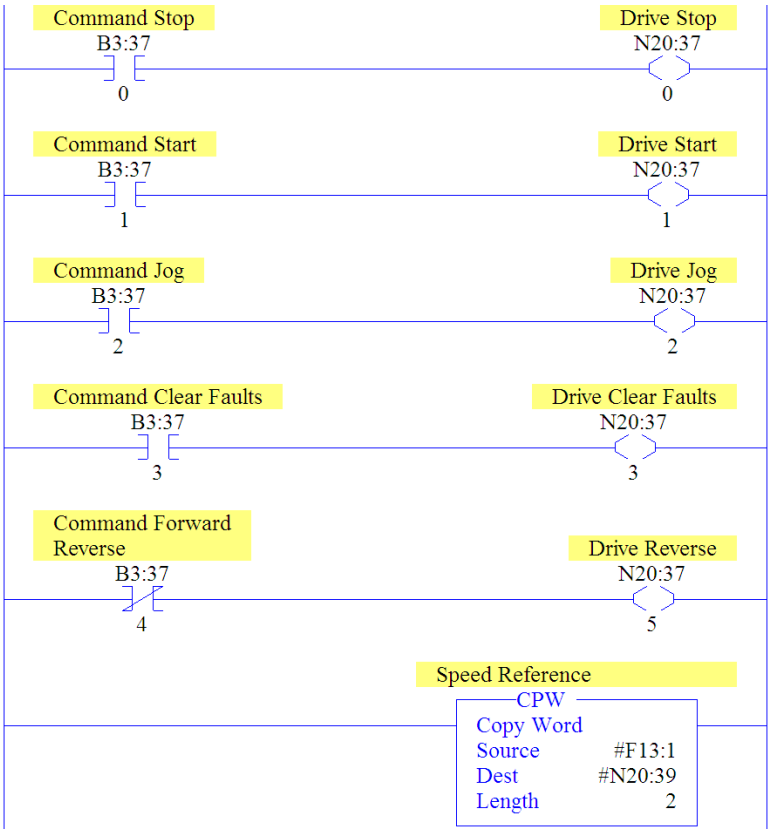
An example ladder logic program that uses these alternate controller data table addresses is shown in Figure 23 and Figure 24.

Figure 23 - PLC-5, SLC 500, and MicroLogix 1100/1400 Controller Example Ladder Logic Program for Logic Status/Feedback



IMPORTANT This ladder does not include logic for Datalinks. However, if Datalinks are required and they are assigned to parameters that are a REAL (floating point) data type, a data conversion must be used. For MicroLogix 1100/1400 controllers only, use a CPW (Copy Word) instruction as shown in the example ladder. For PLC-5 and SLC 500 controllers, use a COP (Copy) instruction.

Figure 24 - PLC-5, SLC 500, and MicroLogix 1100 Controller Example Ladder Logic Program for Logic Command/Reference



IMPORTANT This ladder does not include logic for Datalinks. However, if Datalinks are required and they are assigned to parameters that are a REAL (floating point) data type, a data conversion must be used. For MicroLogix 1100/1400 controllers only, use a CPW (Copy Word) instruction as shown in the example ladder. For PLC-5 and SLC 500 controllers, use a COP (Copy) instruction.

Notes:

Using Explicit Messaging

This chapter provides information and examples that explain how to use Explicit Messaging to configure and monitor the option module and connected PowerFlex 750-Series drive.

ATTENTION: Risk of injury or equipment damage exists. The examples in this publication are intended solely for purposes of example. There are many variables and requirements with any application. Rockwell Automation, Inc. does not assume responsibility or liability (to include intellectual property liability) for actual use of the examples shown in this publication.

ATTENTION: Risk of equipment damage exists. If Explicit Messages are programmed to write parameter data to Nonvolatile Storage (NVS) frequently, the NVS will quickly exceed its life cycle and cause the drive to malfunction. Do not create a program that frequently uses Explicit Messages to write parameter data to NVS. Datalinks do not write to NVS and should be used for frequently changed parameters.

See Chapter 2 for information about the I/O Image, using Logic Command/Status, Reference/Feedback, and Datalinks.

About Explicit Messaging

Explicit Messaging is used to transfer data that does not require continuous updates. With Explicit Messaging, you can configure and monitor a slave device's parameters on the network.

IMPORTANT When an explicit message is performed, by default no connection is made because it is an 'unconnected' message. When timing of the message transaction is important, you can create a dedicated message connection between the controller and drive by checking the 'Connected' box on the Communications tab message configuration dialog box during message setup. These message connections are in addition to the I/O connection. However, the trade off for more message connections is decreased network performance. If your application cannot tolerate this, do not check the 'Connected' box, which is recommended.

TIP To message to another device in a different drive port, see the Instance table in Appendix C:

- DPI Parameter Object section on page 94 for 'Device' parameters.
- Host DPI Parameter Object section on page 108 for 'Host' parameters.

In the Message Configuration dialog box, set the Instance field to an appropriate value within the range listed for the port in which the device resides.

IMPORTANT PowerFlex 750-Series drives have explicit messaging limitations. Table 6 shows the EtherNet/IP Object Class code compatibilities for these drives.

Table 6 - Explicit Messaging Class Code Compatibility with PowerFlex 750-Series Drive

EtherNet/IP Object Class Code	Compatibility	Explicit Messaging Function
Parameter Object 0x0F	No	Single parameter reads/writes
DPI Parameter Object 0x93	Yes (1) with limitations	Single and scattered parameter reads/writes
Host DPI Parameter Object 0x9F	Yes (2) with limitations	Single and scattered parameter reads/writes

(1) Enables access to drive parameters (Port 0), DPI device parameters (Ports 1…6 only), and Host parameters (Ports 7…14 only). For example, DPI Parameter Object Class code 0x93 can access a Safe Speed Monitor option module in Port 6. However, Class code 0x93 cannot access, for example, the Host parameters in a 24V I/O option module in Port 5. See DPI Parameter Object on page 94 for instance (parameter) numbering.

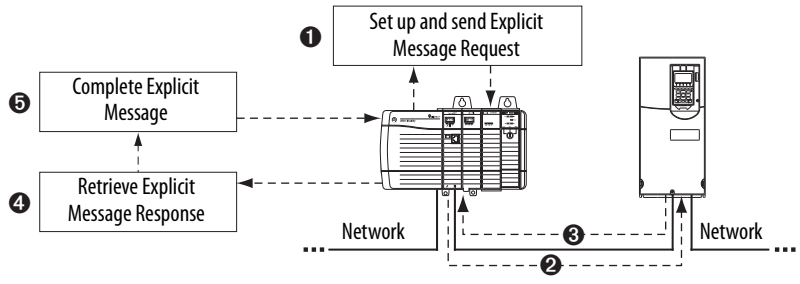
(2) Enables access to drive parameters (Port 0) and Host parameters for all ports (1…14). Host DPI Parameter Object Class code 0x9F cannot access DPI (device) parameters. For example, if a 20-750-DNET option module is in Port 4, its Host parameters can be accessed, but not its DPI (device) parameters. See Host DPI Parameter Object on page 108 for instance (parameter) numbering.

Performing Explicit Messaging

There are five basic events in the Explicit Messaging process. The details of each step will vary depending on the type of controller being used. See the documentation for your controller.

IMPORTANT There must be a request message and a response message for all Explicit Messages, whether you are reading or writing data.

Figure 25 - Explicit Message Process



Event	Description
❶	You format the required data and set up the ladder logic program to send an Explicit Message request to the scanner or bridge module (download).
❷	The scanner or bridge module transmits the Explicit Message Request to the slave device over the network.
❸	The slave device transmits the Explicit Message Response back to the scanner. The data is stored in the scanner buffer.
❹	The controller retrieves the Explicit Message Response from the scanner's buffer (upload).
❺	The Explicit Message is complete.

For information on the maximum number of Explicit Messages that can be executed at a time, see the documentation for the bridge or scanner and/or controller that is being used.

PLC-5 Controller Examples

IMPORTANT The PLC-5 must be Series E (Rev. D.1 or later) to support the MultiHop feature that routes messaging to the drive.

IMPORTANT Due to inherent limitations with the PCCC N-File method, only contiguous multiple parameters can be read or written in one explicit message.

For explicit messaging, the N150 N-Files must be used because they are already mapped to specific parameters in the drive and its connected peripherals. This enables direct access to any parameter.

IMPORTANT Explicit messaging with N150 N-Files is supported by the embedded EtherNet/IP adapter in PowerFlex 755 drives, but is **not** supported by 20-750-ENETR Dual-port EtherNet/IP network communication option modules in PowerFlex 750-Series drives.

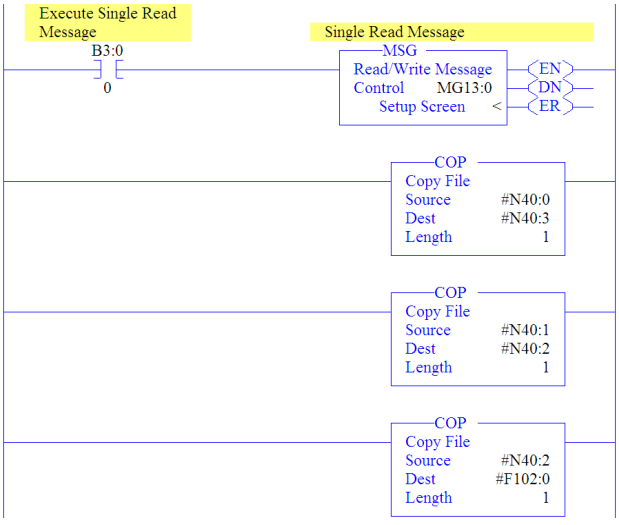
IMPORTANT When performing a write parameter message, the data will always be written to the drive's Nonvolatile Storage (NVS). Over time, continuous writes will exceed the EEPROM life cycle and cause the drive to malfunction.

For PCCC N150 N-File information, see page 90.

PLC-5 Controller Example Ladder Logic Program to Read a Single Parameter

A read message is used to read a single parameter. The specific N150:14 address shown in this read message example reads the value of the 32-bit REAL (floating point) parameter 007 - [Output Current] in a PowerFlex 750-Series drive.

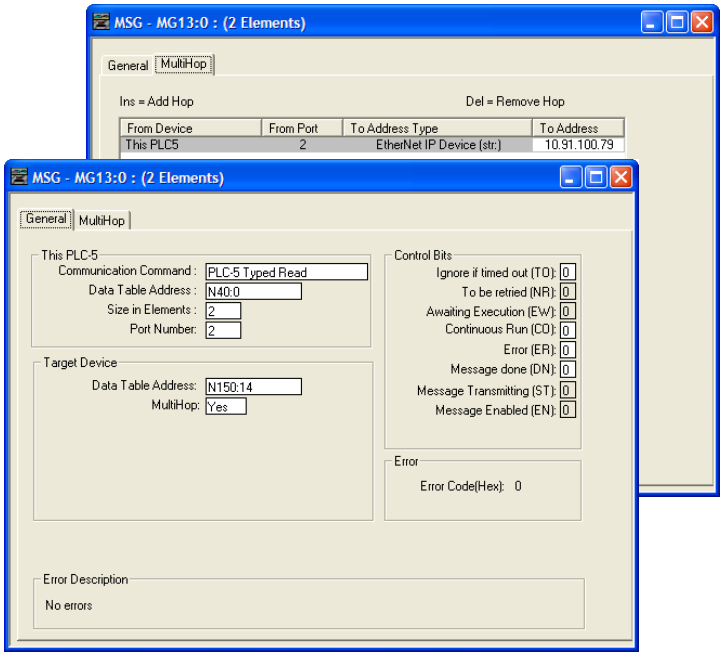
Figure 26 - Example Ladder Logic to Read a Single Parameter



Three COP (Copy) instructions are required to convert the 16-bit integer data table addresses N40:0 (Least Significant Word) and N40:1 (Most Significant Word) to a 32-bit REAL (floating point) data table address F102:0 for correct presentation. The first two COP instructions swap the LSW and MSW, and the third COP instruction correctly presents the 32-bit REAL (floating point) value.

PLC-5 Controller – Formatting a Message to Read a Single Parameter

Figure 27 - Read Single Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to read a single parameter.

General Tab	Example Value	Description
Communication Command	PLC-5 Typed Read	Controller type and command type for controller to read data from the drive.
Data Table Address	N40:0	An unused controller data table address containing the message instruction. This address is the starting word of the destination file.
Size in Elements	2 (1)	Number of elements (words) to be transferred. Each element size is a 16-bit integer.
Port Number	2	Controller port to which the network is connected.
Data Table Address	N150:14 (2)	Specific starting address of the source file in the drive (see page 90).
MultiHop	Yes	Enables communication to allow network messaging to be routed to the drive.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

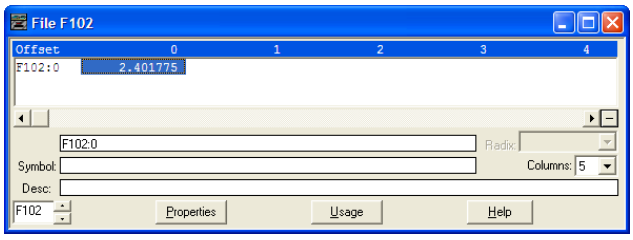
(1) Because the N-Files used for the data transfer occupies two contiguous 16-bit words, the Size in Elements must always be set to 2 regardless of whether the parameter being read is a 16-bit parameter or 32-bit parameter.

(2) See page 90 for N-File addressing.

PLC-5 Controller Example Single Read Response Data

In this specific N150:14 message example, we use the data table address in Figure 28 to store the response value (2.401775 amps) that was read from drive parameter 007 - [Output Current].

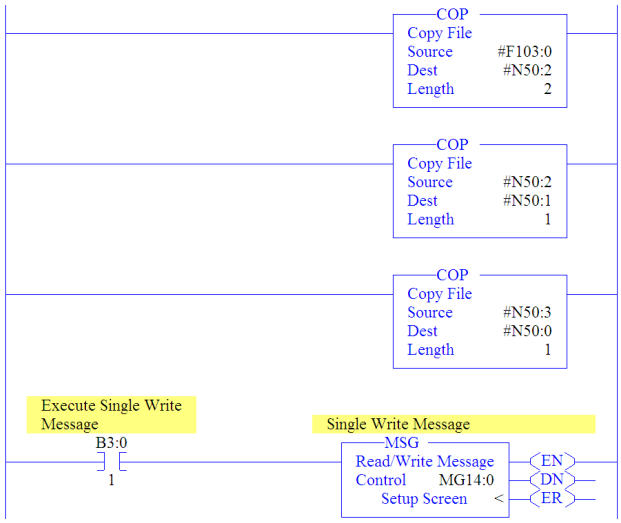
Figure 28 - Example Single Read Response Data File



PLC-5 Controller Example Ladder Logic Program to Write a Single Parameter

A write message is used to write to a single parameter. The specific N154:70 address shown in this write message example writes a value to the 32-bit REAL (floating point) parameter 535 - [Accel Time 1] in a PowerFlex 750-Series drive.

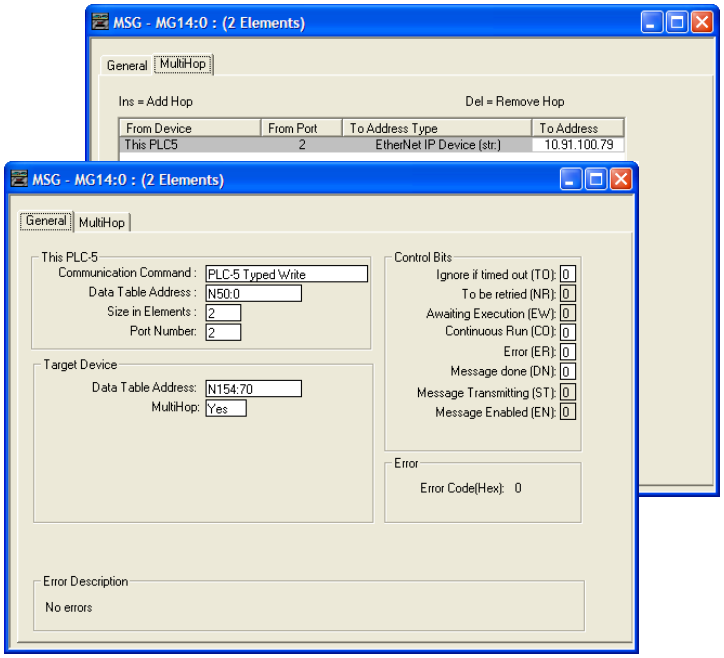
Figure 29 - Example Ladder Logic to Write a Single Parameter



Three COP (Copy) instructions are required to convert the 16-bit integer data table addresses N50:0 (Least Significant Word) and N50:1 (Most Significant Word) to a 32-bit REAL (floating point) data table address F103:0 for correct presentation. The first COP instruction correctly writes the 32-bit REAL (floating point) value. The second and third COP instructions swap the LSW and MSW.

PLC-5 Controller – Formatting a Message to Write a Single Parameter

Figure 30 - Write Single Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to write a single parameter.

General Tab	Example Value	Description
Communication Command	PLC-5 Typed Write (1)	Controller type and command type for controller to write data to the drive.
Data Table Address	N50:0	An unused controller data table address containing the message instruction. This address is the starting word of the source file.
Size in Elements	2 (2)	Number of elements (words) to be transferred. Each element size is a 16-bit integer.
Port Number	2	Controller port to which the network is connected.
Data Table Address	N154:70 (3)	Specific starting address of the destination file in the drive (see page 90).
MultiHop	Yes	Enables communication to allow network messaging to be routed to the drive.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) **Important:** PCCC N150 N-File write messages are written to the drive's EEPROM. Be very cautious as the EEPROM may quickly exceed its life cycle and cause the drive to malfunction.

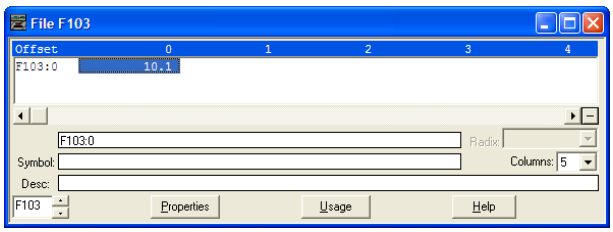
(2) Because the N154 file used for the data transfer occupies two contiguous 16-bit words, the Size in Elements must always be set to 2 regardless of whether the parameter being written to is a 16-bit parameter or 32-bit parameter.

(3) See page 90 for N-File addressing.

PLC-5 Controller Example Single Write Request Data

In this specific N154:70 message example, we use the data table address in Figure 31 to store the request value (10.1 sec.) that was written to drive parameter 535 - [Accel Time 1].

Figure 31 - Example Single Write Request Data File



TIP To verify that the parameter value was successfully written, use the HIM, DriveExplorer software, or DriveExecutive software to access the parameter and view its newly written value.

PLC-5 Controller Reading/Writing Multiple Parameters

You can read or write only contiguous parameters. Scattered read/write messaging is not supported. Also, the range of contiguous parameters must be contained in the same N-File. Two elements (words) are required for each parameter being read or written. For example, to read 5 contiguous parameters, 10 elements (words) must be used.

SLC 500 Controller Examples

When using RSLogix 500 software, version 7.10 or earlier, explicit messaging must be performed using the PCCC N-File method. For RSLogix 500 software, version 7.20 or later, the CIP messaging method has been added along with the PCCC N-File method. However, it is recommended to use the CIP method because it is easier to use and understand. For this reason, only instructions for the CIP method are provided. If you must use the PCCC N-File method, see the PLC-5 Controller Examples on page 52.

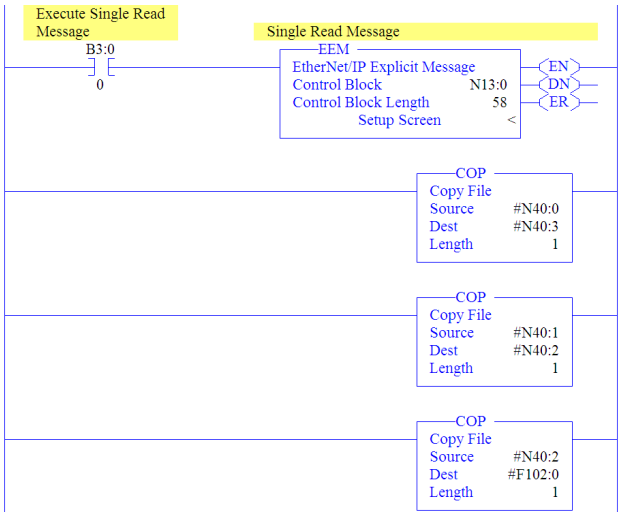
The CIP messaging method provides a Generic Get/Set Attribute Service, which can be used to perform single parameter read or write and multiple parameter read or write explicit messages. Also, the Generic Set Attribute Service offers the choice of writing the data to the drive's Nonvolatile Storage (NVS) or the drive's Random Access Memory (RAM; for Generic Set Attribute Single service only, see page 60). Note that when selecting the data to be written to RAM, the data will be lost if the drive loses power.

For supported classes, instances, and attributes, see Appendix A, EtherNet/IP Objects.

SLC 500 Controller Example Ladder Logic Program to Read a Single Parameter

A Generic Get Attribute Single message is used to read a single parameter. This read message example reads the value of the 32-bit REAL (floating point) parameter 007 - [Output Current] in a PowerFlex 750-Series drive.

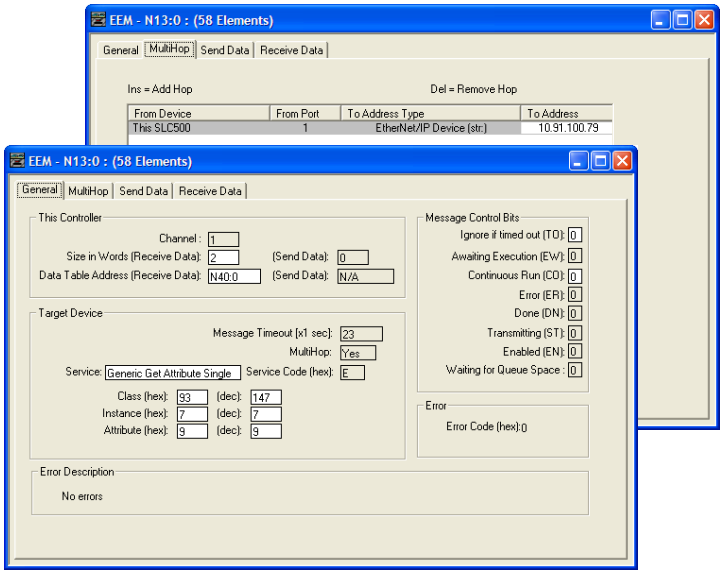
Figure 32 - Example Ladder Logic to Read a Single Parameter



Three COP (Copy) instructions are required to convert the 16-bit integer data table addresses N40:0 (Least Significant Word) and N40:1 (Most Significant Word) to a 32-bit REAL (floating point) data table address F102:0 for correct presentation. The first two COP instructions swap the LSW and MSW, and the third COP instruction correctly presents the 32-bit REAL (floating point) value.

SLC 500 Controller – Formatting a Message to Read a Single Parameter

Figure 33 - Generic Get Attribute Single Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to read a single parameter.

General Tab	Example Value	Description
Size in Words	2 (3)	Number of words to be transferred. Each word size is a 16-bit integer.
Data Table Address	N40:0	An unused controller data table address containing the message instruction. This address is the starting word of the response file.
Service (1)	Generic Get Attribute Single	Code for the requested service.
Class	93 or 9F (Hex.) (4)	Class ID for the DPI Parameter Object.
Instance (2)	7 (Dec.)	Instance number is the same as the parameter number.
Attribute	9 (Dec.)	Attribute number for the Parameter Value attribute.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) The default setting for Service is 'Custom', enabling entry of a Service Code not available from the Service pull-down menu. When choosing a Service other than 'Custom' from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box that is dimmed (unavailable).

(2) The instance is the parameter number in the drive (Port 0). For example, to read parameter 4 of a peripheral in Port 5 of a PowerFlex 755 drive, the instance would be 21504 + 4 = 21508. See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) to determine the instance number.

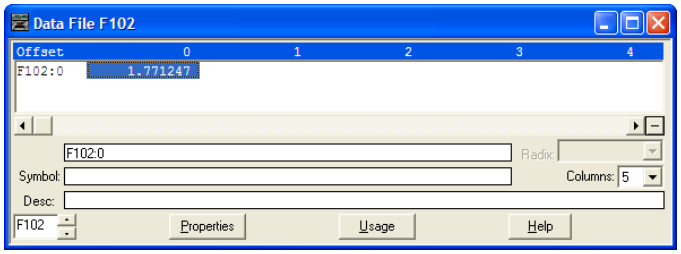
(3) In this example, Output Current is a 32-bit REAL (floating point) parameter. If the parameter being read is a 32-bit integer parameter, the Size in Words would also be set to 2. When the parameter being read is a 16-bit parameter, the Size in Words would be set to 1. See the drive documentation to determine the size of the parameter and its data type.

(4) See Table 6 on page 50 for limitations of PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

SLC 500 Controller Example Get Attribute Single Response Data

In this message example, we use the data table address in Figure 34 to store the response value (1.771247 amps) that was read from drive parameter 007 - [Output Current].

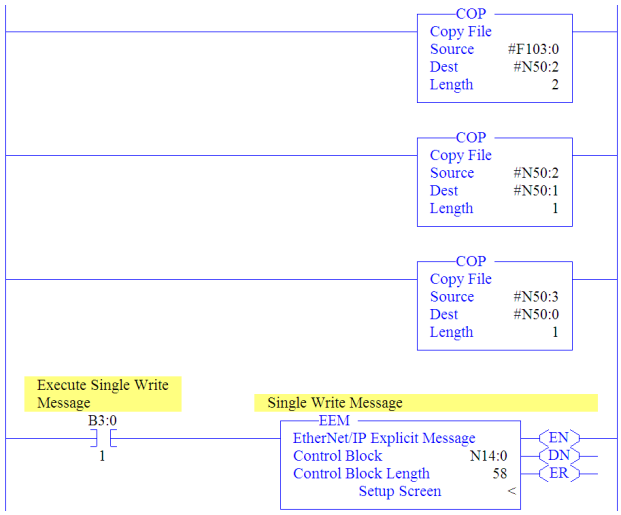
Figure 34 - Example Get Attribute Single Response Data File



SLC 500 Controller Example Ladder Logic Program to Write a Single Parameter

A Generic Set Attribute Single message is used to write to a single parameter. This write message example writes a value to the 32-bit REAL (floating point) parameter 535 - [Accel Time 1] in a PowerFlex 750-Series drive.

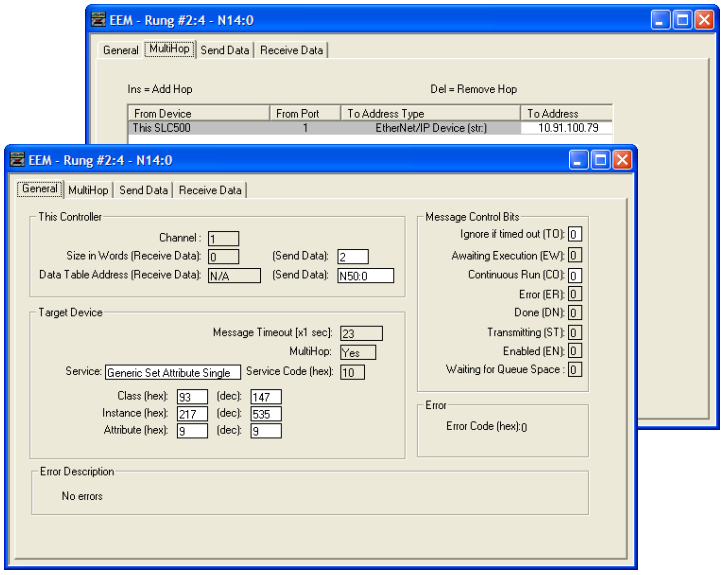
Figure 35 - Example Ladder Logic to Write a Single Parameter



Three COP (Copy) instructions are required to convert the 16-bit integer data table addresses N50:0 (Least Significant Word) and N50:1 (Most Significant Word) to a 32-bit REAL (floating point) data table address F103:0 for correct presentation. The first COP instruction correctly writes the 32-bit REAL (floating point) value. The second and third COP instructions swap the LSW and MSW.

SLC 500 Controller – Formatting a Message to Write a Single Parameter

Figure 36 - Generic Set Attribute Single Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to write a single parameter.

General Tab	Example Value	Description
Size in Words	2 (4)	Number of words to be transferred. Each word size is a 16-bit integer.
Data Table Address	N50:0	An unused controller data table address containing the message instruction. This address is the starting word of the request file.
Service (1)	Generic Set Attribute Single	Code for the requested service.
Class	93 or 9F (Hex.) (5)	Class ID for the DPI Parameter Object.
Instance (2)	535 (Dec.)	Instance number is the same as the parameter number.
Attribute (3)	9 or 10 (Dec.)	Attribute number for the Parameter Value attribute.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) The default setting for Service is 'Custom', enabling entry of a Service Code not available from the Service pull-down menu. When choosing a Service other than 'Custom' from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box that is dimmed (unavailable).

(2) The instance is the parameter number in the drive (Port 0). For example, to read parameter 4 of a peripheral in Port 5 of a PowerFlex 755 drive, the instance would be 21504 + 4 = 21508. See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) to determine the instance number.

(3) Setting the Attribute value to '9' will write the parameter value to the drive's Nonvolatile Storage (EEPROM) memory, so the parameter value will remain even after the drive is power cycled. **Important:** When set to '9', be very cautious as the EEPROM may quickly exceed its life cycle and cause the drive to malfunction. Setting the Attribute value to 'A' (10 decimal) will write the parameter value to temporary memory, so the parameter value will be lost after the drive is power cycled. It is recommended to use the 'A' (10 decimal) setting when frequent write messages are required.

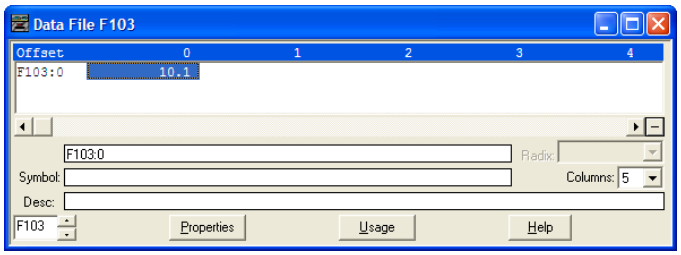
(4) In this example, Accel Time 1 is a 32-bit REAL (floating point) parameter. If the parameter being written to is a 32-bit integer parameter, the Size in Words would also be set to 2. When the parameter being written to is a 16-bit parameter, the Size in Words would be set to 1. See the drive documentation to determine the size of the parameter and its data type.

(5) See Table 6 on page 50 for limitations of PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

SLC 500 Controller Example Set Attribute Single Request Data

In this message example, we use the data table address in Figure 37 to store the request value (10.1 sec.) that was written to drive parameter 535 - [Accel Time 1].

Figure 37 - Example Set Attribute Single Request Data File



TIP To verify that the parameter value was successfully written, use the HIM, DriveExplorer software, or DriveExecutive software to access the parameter and view its newly written value.

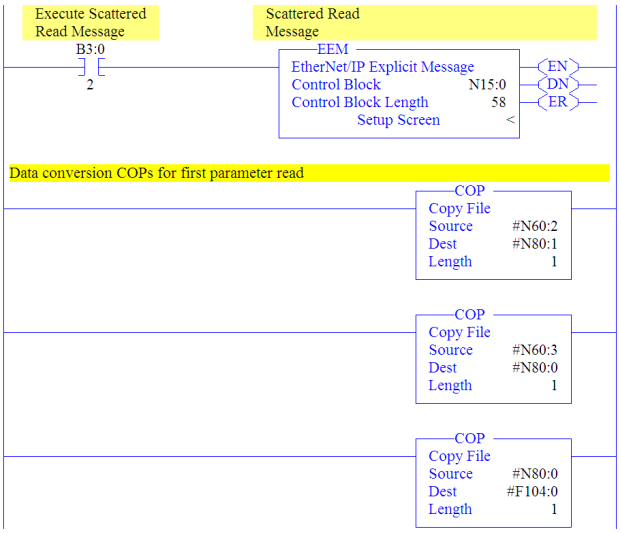
SLC 500 Controller Example Ladder Logic Program to Read Multiple Parameters

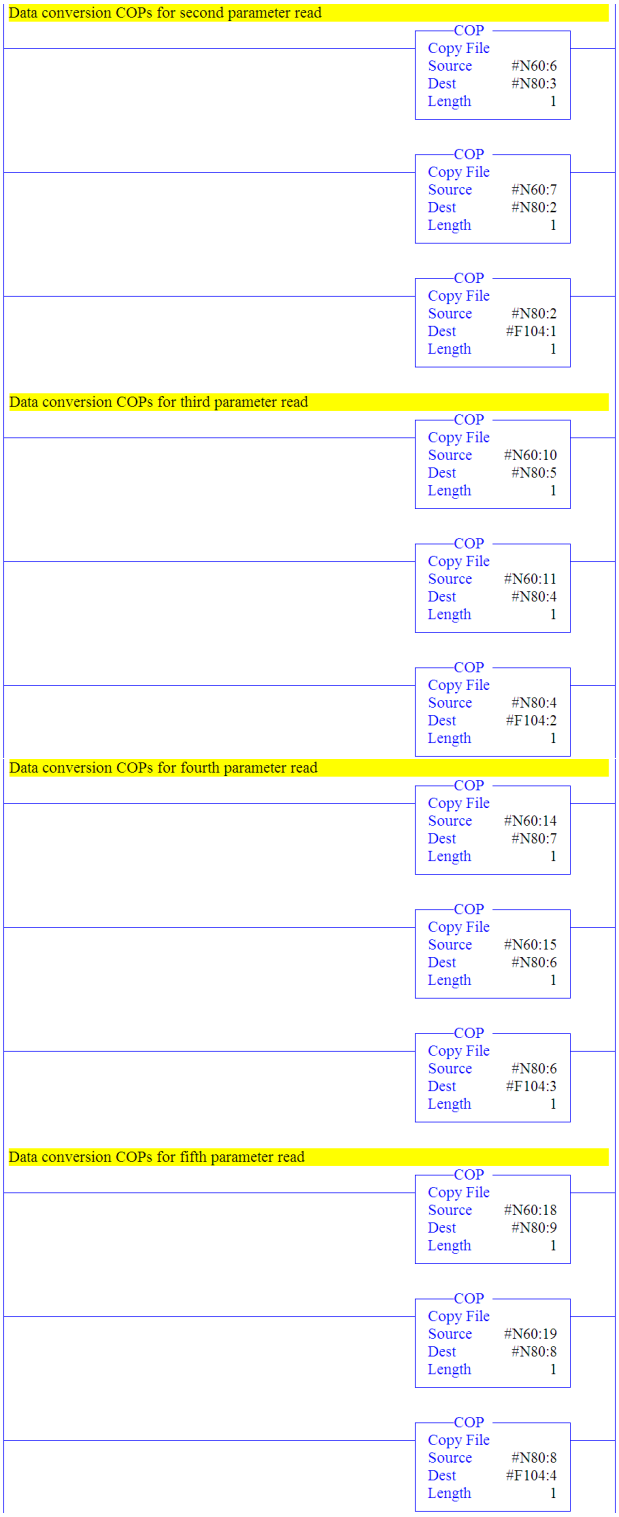
A Custom scattered read message is used to read the values of multiple parameters. This read message example reads the values of these five 32-bit REAL (floating point) parameters in a PowerFlex 750-Series drive:

- Parameter 001 - [Output Frequency]
- Parameter 007 - [Output Current]
- Parameter 008 - [Output Voltage]
- Parameter 009 - [Output Power]
- Parameter 011 - [DC Bus Volts]

See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) for parameter numbering.

Figure 38 - Example Ladder Logic to Read Multiple Parameters

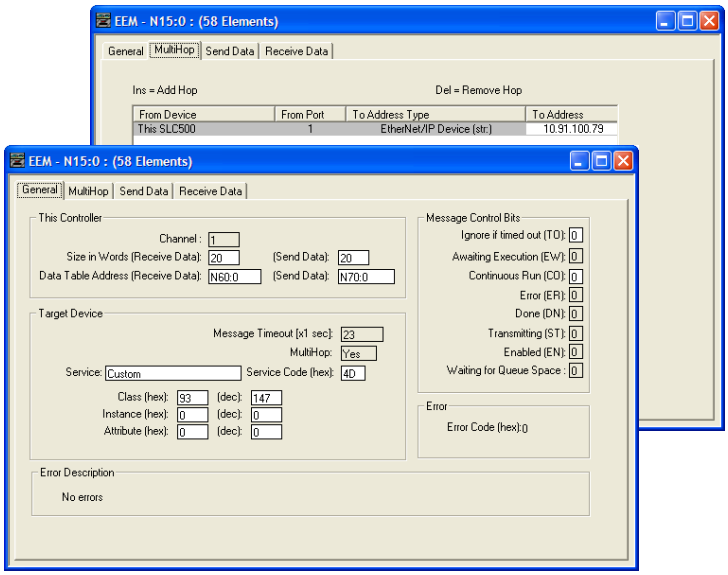




For each parameter being read, three COP (Copy) instructions are required to convert the 16-bit integer data table addresses, for example N60:2 (Least Significant Word) and N60:3 (Most Significant Word) for the first parameter, to a 32-bit REAL (floating point) data table address F104:0 for correct presentation. The first two COP instructions swap the LSW and MSW, and the third COP instruction correctly presents the 32-bit REAL (floating point) value.

SLC 500 Controller – Formatting a Message to Read Multiple Parameters

Figure 39 - Custom Scattered Read Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to read multiple parameters.

General Tab	Example Value	Description
Size in Words		Each word size is a 16-bit integer.
Receive Data	20 (2)	Number of words to be received.
Send Data	20 (2)	Number of words to be sent.
Data Table Address		An unused controller data table address containing the message instruction.
Receive Data	N60:0	This address is the starting word of the response file.
Send Data	N70:0	This address is the starting word of the request file.
Service (1)	Custom	Required for scattered messages.
Service Code	4D (Hex.)	Code for the requested service.
Class	93 or 9F (Hex.) (3)	Class ID for the DPI Parameter Object.
Instance	0 (Dec.)	Required for scattered messages.
Attribute	0 (Dec.)	Required for scattered messages.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) The default setting for Service is 'Custom', enabling entry of a Service Code not available from the Service pull-down menu. When choosing a Service other than 'Custom' from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box that is dimmed (unavailable).

(2) In this example, we are reading five 32-bit REAL (floating point) parameters. Each parameter being read requires four contiguous 16-bit words. Scattered read messages always assume that every parameter being read is a 32-bit parameter, regardless of its actual size. The data structure format is shown on page 70. Maximum length is 128 words, which equates to 32 parameters. For parameter numbering, see DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F).

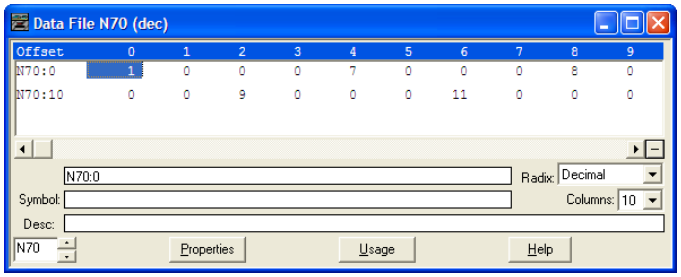
(3) See Table 6 on page 50 for limitations of PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

SLC 500 Controller Example Scattered Read Request Data

In this message example, we use the data table addresses in Figure 40 to store the request values to be read from these five 32-bit REAL (floating point) parameters in a PowerFlex 750-Series drive:

- Parameter 001 - [Output Frequency]
- Parameter 007 - [Output Current]
- Parameter 008 - [Output Voltage]
- Parameter 009 - [Output Power]
- Parameter 011 - [DC Bus Volts]

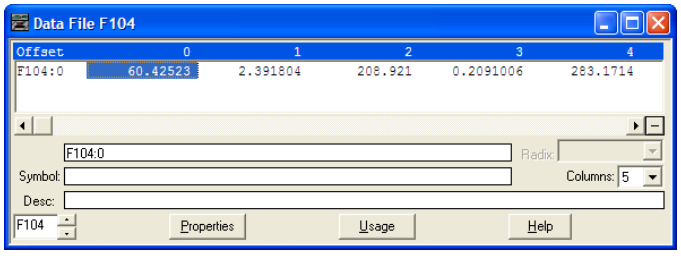
Figure 40 - Example Scattered Read Request Data File



SLC 500 Controller Example Scattered Read Response Data

In this message example, we use the data table addresses in Figure 41 to store the response values that were read from the requested drive parameters. These values have been converted using a CPW (Copy Word) instruction for correct presentation.

Figure 41 - Example Scattered Read Response Data File



In this example, the parameters have the following values:

PowerFlex 750-Series Drive Parameter	Address	Read Value
1 - [Output Frequency]	F104:0	60.42523 Hz
7 - [Output Current]	F104:1	2.391804 Amp
8 - [Output Voltage]	F104:2	208.921V AC
9 - [Output Power]	F104:3	0.2091006 kW
11 - [DC Bus Voltage]	F104:4	283.1714V DC

SLC 500 Controller Example Ladder Logic Program to Write Multiple Parameters

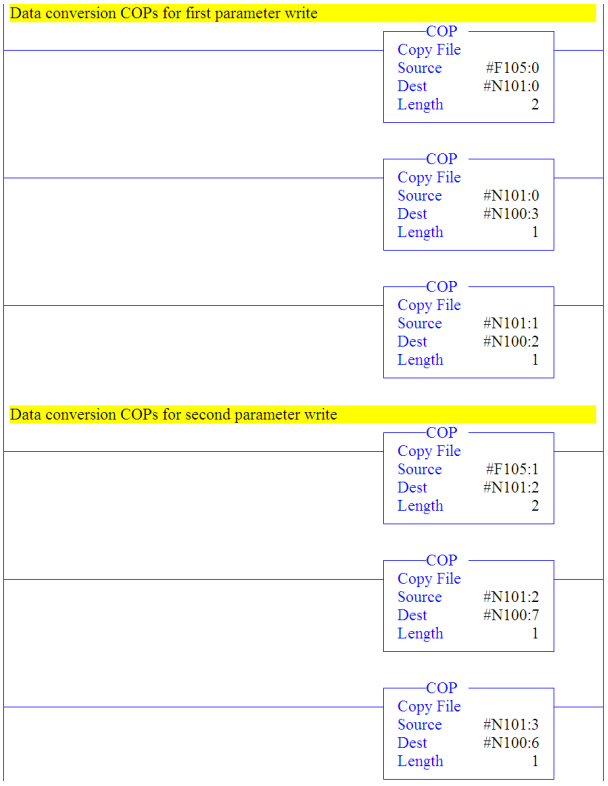
A Custom scattered write message is used to write to multiple parameters. This write message example writes the following values to these five 32-bit REAL (floating point) parameters in a PowerFlex 750-Series drive:

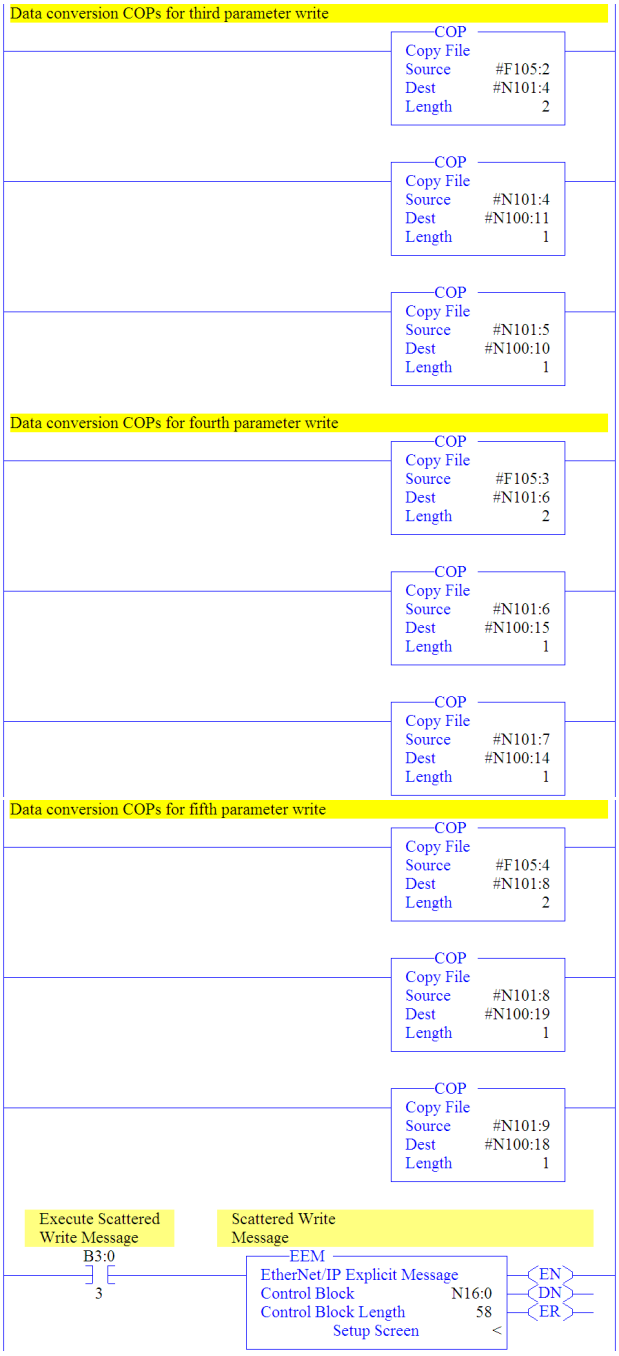
PowerFlex 750-Series Drive Parameter	Write Value
536 - [Accel Time 2]	11.1 Sec
538 - [Decel Time 2]	22.2 Sec
575 - [Preset Speed 5]	33.3 Hz
576 - [Preset Speed 6]	44.4 Hz
577 - [Preset Speed 7]	55.5 Hz

See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) for parameter numbering.

IMPORTANT If the explicit message scattered write must be written continuously, then use a separate Generic Set service explicit message single write for each parameter using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F and attribute A (10 decimal; see page 60). Attribute A writes to RAM—not NVS (EEPROM) memory. This example scattered write message using attribute 0 writes to NVS. Over time, continuous writes will exceed the EEPROM life cycle and cause the drive to malfunction.

Figure 42 - Example Ladder Logic to Write Multiple Parameters

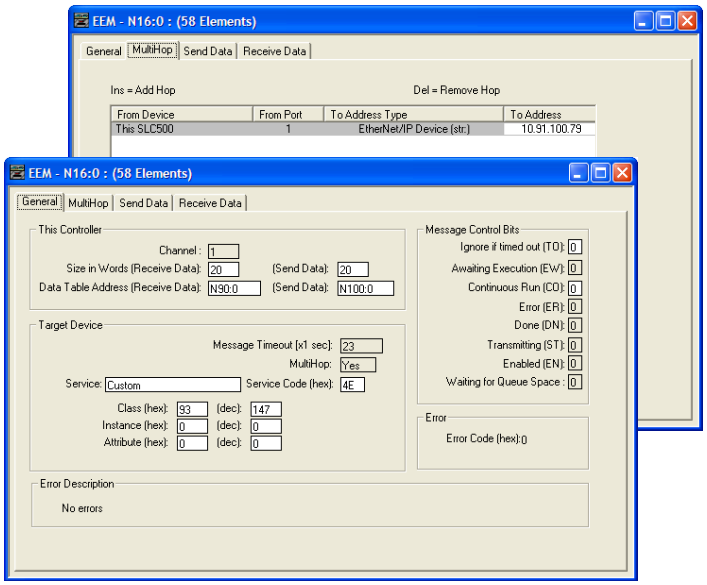




For each parameter being written to, three COP (Copy) instructions are required to convert the 16-bit integer data table addresses, for example N100:2 (Least Significant Word) and N100:3 (Most Significant Word) for the first parameter, to a 32-bit REAL (floating point) data table address F105:0 for correct presentation. The first COP instruction correctly writes the 32-bit REAL (floating point) value. The second and third COP instructions swap the LSW and MSW.

SLC 500 Controller – Formatting a Message to Write Multiple Parameters

Figure 43 - Custom Scattered Write Multiple Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to write multiple parameters.

General Tab	Example Value	Description
Size in Words		Each word size is a 16-bit integer.
Receive Data	20 (2)	Number of words to be received.
Send Data	20 (2)	Number of words to be sent.
Data Table Address		An unused controller data table address containing the message instruction.
Receive Data	N90:0	This address is the starting word of the response file.
Send Data	N100:0	This address is the starting word of the request file.
Service (1)	Custom	Required for scattered messages.
Service Code	4E (Hex.) (3)	Code for the requested service.
Class	93 or 9F (Hex.) (4)	Class ID for the DPI Parameter Object.
Instance	0 (Dec.)	Required for scattered messages.
Attribute	0 (Dec.)	Required for scattered messages.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) The default setting for Service is 'Custom', enabling entry of a Service Code not available from the Service pull-down menu. When choosing a Service other than 'Custom' from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box that is dimmed (unavailable).

(2) In this example, we are writing to five 32-bit REAL (floating point) parameters. Each parameter being written to requires four contiguous 16-bit words. Scattered write messages always assume that every parameter being written to is a 32-bit parameter, regardless of its actual size. The data structure format is shown on page 70. Maximum length is 128 words, which equates to 32 parameters. For parameter numbering, see DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F).

(3) Service Code 4E write messages are written to the drive's Nonvolatile Storage (EEPROM) memory, so the parameter value will remain even after the drive is power cycled. **Important:** Be very cautious as the EEPROM may quickly exceed its life cycle and cause the drive to malfunction.

(4) See Table 6 on page 50 for limitations of PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

SLC 500 Controller Example Scattered Write Request Data

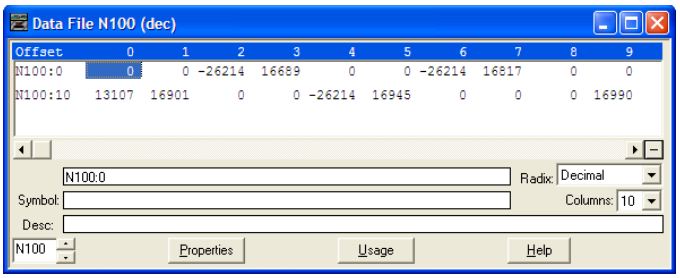
In this message example, we use the F105: data table addresses to store the request values to be written to these 32-bit REAL (floating point) parameters:

PowerFlex 750-Series Drive Parameter	Address	Write Value
536 - [Accel Time 2]	F105:0	11.1 Sec
538 - [Decel Time 2]	F105:1	22.2 Sec
575 - [Preset Speed 5]	F105:2	33.3 Hz
576 - [Preset Speed 6]	F105:3	44.4 Hz
577 - [Preset Speed 7]	F105:4	55.5 Hz

See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) for parameter numbering.

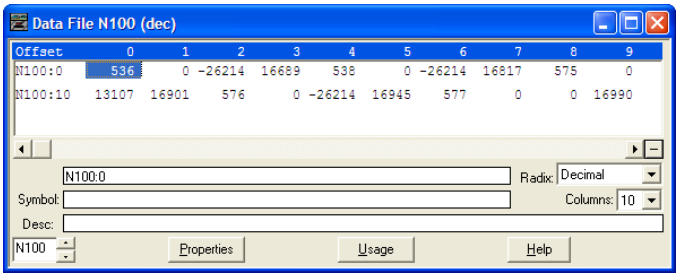
Figure 44 shows the parameter values which, in this example, have been converted using a CPW (Copy Word) instruction—one for each value—to correctly write their values. The CPW instruction separates the 32-bit REAL (floating point) value (for example, F105:0 that contains 11.1 seconds) into two 16-bit integers (for example N100:2 and N100:3). While the values in the 16-bit integer registers represent the actual values being written in the 32-bit floating point registers, they will not appear correct, but the message will properly decode them.

Figure 44 - Example Scattered Write Request Unconverted Data File



To complete message configuration, the numbers of the parameters being written to must now be entered in the appropriate N100: data table registers as shown in Figure 45 for this example.

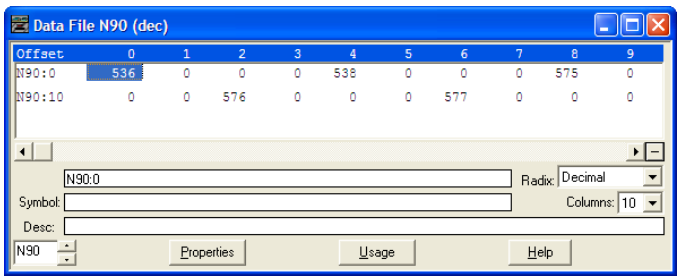
Figure 45 - Example Scattered Write Request Unconverted Data File with Entered Parameters



SLC 500 Controller Example Scattered Write Response Data

In this message example, we use the data table addresses in Figure 46 to store the response values that were written to the requested drive parameters. Values of '0' indicate no errors occurred.

Figure 46 - Example Scattered Write Response Data File



TIP To verify that the parameter values were successfully written, use the HIM, DriveExplorer software, or DriveExecutive software to access the parameters and view their newly written values.

SLC 500 Controller – Explanation of Request and Response Data for Read/Write Multiple Messaging

The data structures in Figure 47 use 32-bit words and can accommodate up to 32 parameters in a single message. In the Response Message, a parameter number with Bit 15 set indicates that the associated parameter value field contains an error code (parameter number in response data will be negative).

Figure 47 - Data Structures for Scattered Read/Write Messages

	Request (Source Data)		Response (Destination Data)
Word 0	Parameter Number (LSW)	Word 0	Parameter Number (LSW)
1	Parameter Number (MSW)	1	Parameter Number (MSW)
2	Parameter Value (LSW)	2	Parameter Value (LSW)
3	Parameter Value (MSW)	3	Parameter Value (MSW)
4	Parameter Number (LSW)	4	Parameter Number (LSW)
5	Parameter Number (MSW)	5	Parameter Number (MSW)
6	Parameter Value (LSW)	6	Parameter Value (LSW)
7	Parameter Value (MSW)	7	Parameter Value (MSW)
8	Parameter Number (LSW)	8	Parameter Number (LSW)
9	Parameter Number (MSW)	9	Parameter Number (MSW)
10	Parameter Value (LSW)	10	Parameter Value (LSW)
11	Parameter Value (MSW)	11	Parameter Value (MSW)
12	Parameter Number (LSW)	12	Parameter Number (LSW)
13	Parameter Number (MSW)	13	Parameter Number (MSW)
14	Parameter Value (LSW)	14	Parameter Value (LSW)
15	Parameter Value (MSW)	15	Parameter Value (MSW)
16	Parameter Number (LSW)	16	Parameter Number (LSW)
17	Parameter Number (MSW)	17	Parameter Number (MSW)
18	Parameter Value (LSW)	18	Parameter Value (LSW)
19	Parameter Value (MSW)	19	Parameter Value (MSW)
20	Parameter Number (LSW)	20	Parameter Number (LSW)
21	Parameter Number (MSW)	21	Parameter Number (MSW)
22	Parameter Value (LSW)	22	Parameter Value (LSW)
23	Parameter Value (MSW)	23	Parameter Value (MSW)
24	Parameter Number (LSW)	24	Parameter Number (LSW)
25	Parameter Number (MSW)	25	Parameter Number (MSW)
26	Parameter Value (LSW)	26	Parameter Value (LSW)
27	Parameter Value (MSW)	27	Parameter Value (MSW)
28	Parameter Number (LSW)	28	Parameter Number (LSW)
29	Parameter Number (MSW)	29	Parameter Number (MSW)
30	Parameter Value (LSW)	30	Parameter Value (LSW)
31	Parameter Value (MSW)	31	Parameter Value (MSW)
32	Parameter Number (LSW)	32	Parameter Number (LSW)
33	Parameter Number (MSW)	33	Parameter Number (MSW)
34	Parameter Value (LSW)	34	Parameter Value (LSW)
35	Parameter Value (MSW)	35	Parameter Value (MSW)
⋮		⋮	
124	Parameter Number (LSW)	124	Parameter Number (LSW)
125	Parameter Number (MSW)	125	Parameter Number (MSW)
126	Parameter Value (LSW)	126	Parameter Value (LSW)
127	Parameter Value (MSW)	127	Parameter Value (MSW)

MicroLogix 1100/1400 Controller Examples

When using RSLogix 500 software, version 7.10 or earlier, explicit messaging must be performed using the PCCC N-File method. For RSLogix 500 software, version 7.20 or later, the CIP messaging method has been added along with the PCCC N-File method. However, the CIP method is recommended because it is easier to use and understand. For this reason, only instructions for the CIP method are provided.

IMPORTANT Due to inherent limitations with the PCCC N-File method, only contiguous multiple parameters can be read or written in one explicit message.

For explicit messaging using the PCCC N-File method, the N150 N-Files must be used because they are already mapped to specific parameters in the drive and its connected peripherals. This enables direct access to any parameter.

For PCCC N150 N-File information, see page 90.

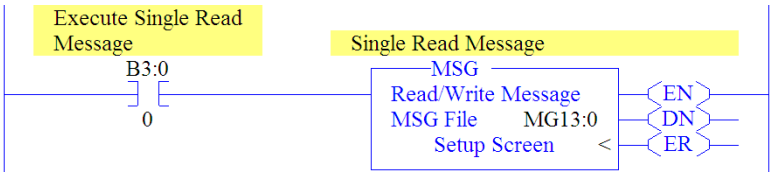
The CIP messaging method provides a Generic Get/Set Attribute Service, which can be used to perform single parameter read or write and multiple parameter read or write explicit messages. Also, the Generic Set Attribute Service offers the choice of writing the data to the drive's Nonvolatile Storage (NVS) or the drive's Random Access Memory (RAM; for Generic Set Attribute Single service only, see page 74). Note that when selecting the data to be written to RAM, the data will be lost if the drive loses power.

For supported classes, instances, and attributes, see Appendix A, EtherNet/IP Objects.

MicroLogix 1100/1400 Controller Example Ladder Logic Program to Read a Single Parameter

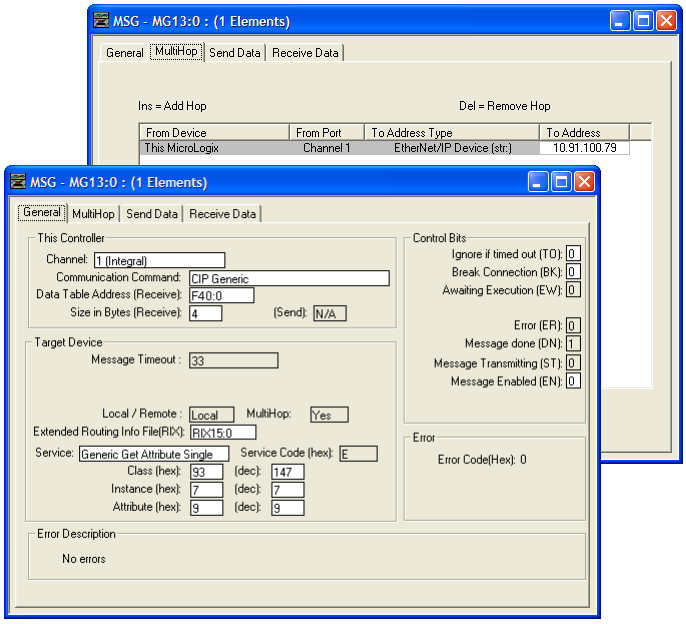
A Generic Get Attribute Single message is used to read a single parameter. This read message example reads the value of the 32-bit REAL (floating point) parameter 007 - [Output Current] in a PowerFlex 750-Series drive.

Figure 48 - Example Ladder Logic to Read a Single Parameter



MicroLogix 1100/1400 Controller – Formatting a Message to Read a Single Parameter

Figure 49 - Generic Get Attribute Single Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to read a single parameter.

General Tab	Example Value	Description
Channel	1	Controller port to which the network is connected.
Comm… Command	CIP Generic	Used to access the DPI Parameter Object in the option module.
Data Table Address	F40:0 (3)	An unused controller data table address containing the message instruction. This address is the starting word of the response file.
Size in Bytes	4 (4)	Number of bytes to be transferred. Each byte size is an 8-bit integer.
Extended Routing…	RIX15:0	An unused routing information file for the controller.
Service (1)	Generic Get Attribute Single	Code for the requested service.
Class	93 or 9F (Hex.) (5)	Class ID for the DPI Parameter Object.
Instance (2)	7 (Dec.)	Instance number is the same as the parameter number.
Attribute	9 (Dec.)	Attribute number for the Parameter Value attribute.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) The default setting for Service is 'Custom', enabling entry of a Service Code not available from the Service pull-down menu. When selecting a Service other than 'Custom' from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box that is dimmed (unavailable).

(2) The instance is the parameter number in the drive (Port 0). For example, to read parameter 4 of a peripheral in Port 5 of a PowerFlex 755 drive, the instance would be 21504 + 4 = 21508. See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) to determine the instance number.

(3) In this example, Output Current is a 32-bit REAL (floating point) parameter. Therefore, set the Data Table Address to 'F' type (floating point). If the parameter being read is a 32-bit integer, the Data Table Address type would be set to 'L' (long word).

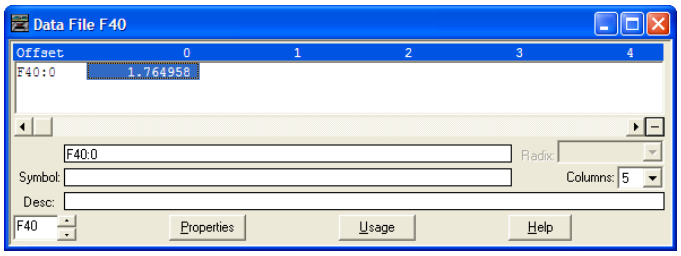
(4) In this example, Output Current is a 32-bit REAL (floating point) parameter. If the parameter being read is a 32-bit integer parameter, the Size in Bytes would also be set to 4. When the parameter being read is a 16-bit parameter, the Size in Bytes would be set to 2. See the drive documentation to determine the size of the parameter and its data type.

(5) See Table 6 on page 50 for limitations of PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

MicroLogix 1100/1400 Controller Example Get Attribute Single Response Data

In this message example, we use the data table address in Figure 50 to store the response value (1.77 amps) that was read from drive parameter 007 - [Output Current].

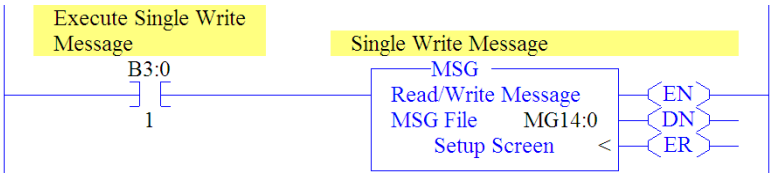
Figure 50 - Example Get Attribute Single Response Data File



MicroLogix 1100/1400 Controller Example Ladder Logic Program to Write a Single Parameter

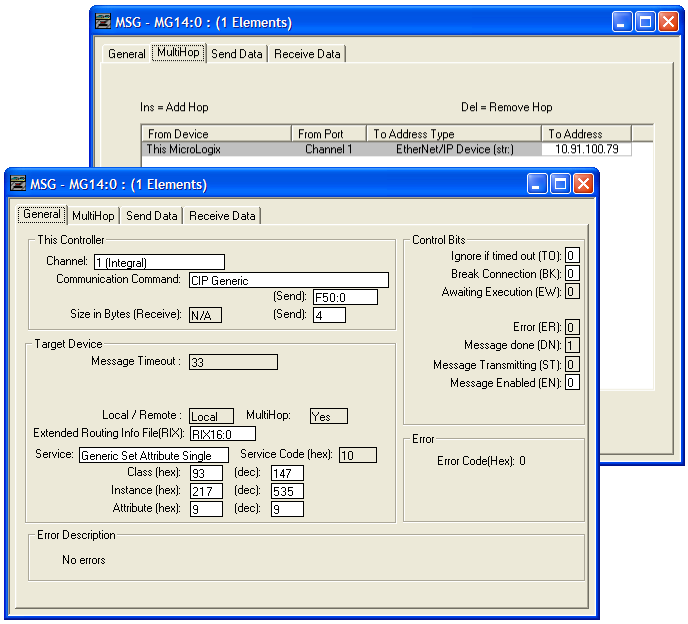
A Generic Set Attribute Single message is used to write to a single parameter. This write message example writes a value to the 32-bit REAL (floating point) parameter 535 - [Accel Time 1] in a PowerFlex 750-Series drive.

Figure 51 - Example Ladder Logic to Write a Single Parameter



MicroLogix 1100/1400 Controller – Formatting a Message to Write a Single Parameter

Figure 52 - Generic Set Attribute Single Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to write a single parameter.

General Tab	Example Value	Description
Channel	1	Controller port to which the network is connected.
Comm… Command	CIP Generic	Used to access the DPI Parameter Object in the option module.
Data Table Address	F50:0 (4)	An unused controller data table address containing the message instruction. This address is the starting word of the request file.
Size in Bytes	4 (5)	Number of bytes to be transferred. Each byte size is an 8-bit integer.
Extended Routing…	RIX16:0	An unused routing information file for the controller.
Service (1)	Generic Set Attribute Single	Code for the requested service.
Class	93 or 9F (Hex.) (6)	Class ID for the DPI Parameter Object.
Instance (2)	535 (Dec.)	Instance number is the same as the parameter number.
Attribute (3)	9 or 10 (Dec.)	Attribute number for the Parameter Value attribute.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) The default setting for Service is 'Custom', enabling entry of a Service Code not available from the Service pull-down menu. When selecting a Service other than 'Custom' from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box that is dimmed (unavailable).

(2) The instance is the parameter number in the drive (Port 0). For example, to read parameter 4 of a peripheral in Port 5 of a PowerFlex 755 drive, the instance would be 21504 + 4 = 21508. See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) to determine the instance number.

(3) Setting the Attribute value to '9' will write the parameter value to the drive's Nonvolatile Storage (EEPROM) memory, so the parameter value will remain even after the drive is power cycled. **Important:** When set to '9', be very cautious as the EEPROM may quickly exceed its life cycle and cause the drive to malfunction. Setting the Attribute value to '10' will write the parameter value to temporary memory, so the parameter value will be lost after the drive is power cycled. It is recommended to use the '10' setting when frequent write messages are required.

(4) In this example, Accel Time 1 is a 32-bit REAL (floating point) parameter. Therefore, set the Data Table Address to 'F' type (floating point). If the parameter being written to is a 32-bit integer, the Data Table Address type would be set to 'L' (long word).

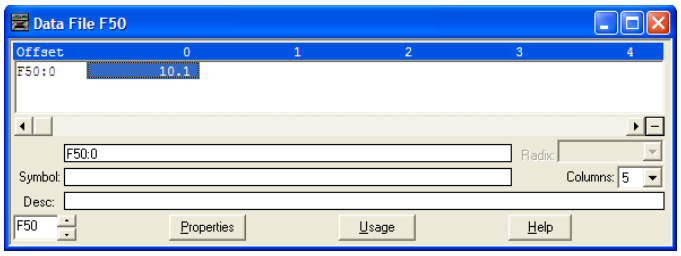
(5) In this example, Accel Time 1 is a 32-bit REAL (floating point) parameter. If the parameter being written to is a 32-bit integer parameter, the Size in Bytes would also be set to 4. When the parameter being written to is a 16-bit parameter, the Size in Bytes would be set to 2.

(6) See Table 6 on page 50 for limitations of PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

MicroLogix 1100/1400 Controller Example Set Attribute Single Request Data

In this message example, we use the data table address in Figure 53 to store the request value (10.1 sec.) that was written to drive parameter 535 - [Accel Time 1].

Figure 53 - Example Set Attribute Single Request Data File



TIP To verify that the parameter value was successfully written, use the HIM, DriveExplorer software, or DriveExecutive software to access the parameter and view its newly written value.

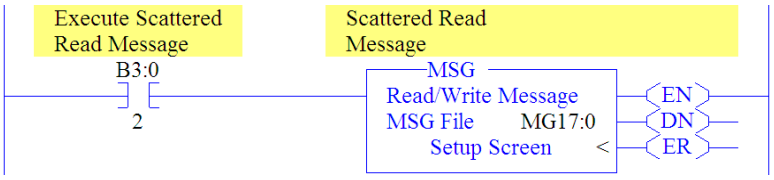
MicroLogix 1100/1400 Controller Example Ladder Logic Program to Read Multiple Parameters

A Custom scattered read message is used to read the values of multiple parameters. This read message example reads the values of these five 32-bit REAL (floating point) parameters in a PowerFlex 750-Series drive:

- Parameter 001 - [Output Frequency]
- Parameter 007 - [Output Current]
- Parameter 008 - [Output Voltage]
- Parameter 009 - [Output Power]
- Parameter 011 - [DC Bus Volts]

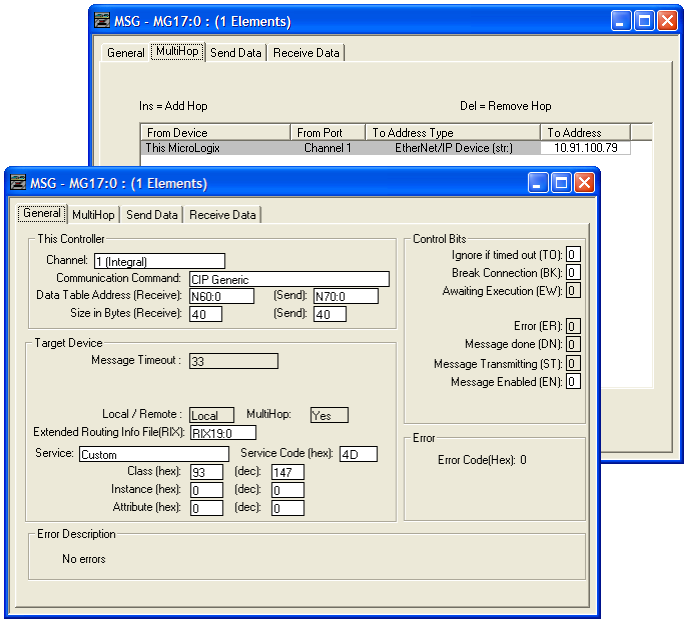
See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) for parameter numbering.

Figure 54 - Example Ladder Logic to Read Multiple Parameters



MicroLogix 1100/1400 Controller – Formatting a Message to Read Multiple Parameters

Figure 55 - Custom Scattered Read Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to read multiple parameters.

General Tab	Example Value	Description
Channel	1	Controller port to which the network is connected.
Comm… Command	CIP Generic	Used to access the DPI Parameter Object in the option module.
Data Table Address		An unused controller data table address containing the message instruction.
Receive	N60:0	This address is the starting word of the response file.
Send	N70:0	This address is the starting word of the request file.
Size in Bytes		Each byte size is an 8-bit integer.
Receive	40 (2)	Number of bytes to be received.
Send	40 (2)	Number of bytes to be sent.
Extended Routing…	RIX19:0	An unused routing information file for the controller.
Service (1)	Custom	Required for scattered messages.
Service Code	4D (Hex.)	Code for the requested service.
Class	93 or 9F (Hex.) (3)	Class ID for the DPI Parameter Object.
Instance	0 (Dec.)	Required for scattered messages.
Attribute	0 (Dec.)	Required for scattered messages.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) The default setting for Service is 'Custom', enabling entry of a Service Code not available from the Service pull-down menu. When choosing a Service other than 'Custom' from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box that is dimmed (unavailable).

(2) In this example, we are reading five 32-bit REAL (floating point) parameters. Each parameter being read requires four contiguous 16-bit words. Scattered read messages always assume that every parameter being read is a 32-bit parameter, regardless of its actual size. Therefore, the Size in Bytes must be set to 40. The data structure format is shown on page 82. Maximum message length is 128 words (256 bytes), which equates to 32 parameters. For parameter numbering, see DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F).

(3) See Table 6 on page 50 for limitations of PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

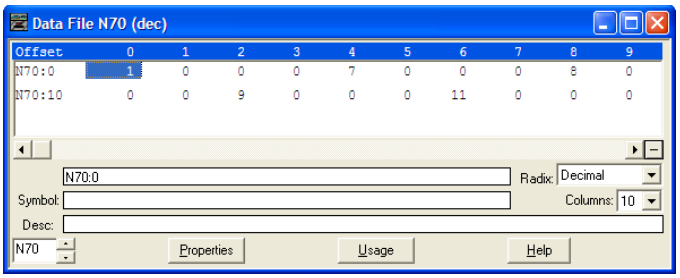
MicroLogix 1100/1400 Controller Example Scattered Read Request Data

In this message example, we use the data table addresses in Figure 56 to store the request values to be read from these five 32-bit REAL (floating point) parameters in a PowerFlex 750-Series drive:

- Parameter 001 - [Output Frequency]
- Parameter 007 - [Output Current]
- Parameter 008 - [Output Voltage]
- Parameter 009 - [Output Power]
- Parameter 011 - [DC Bus Volts]

See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) for parameter numbering.

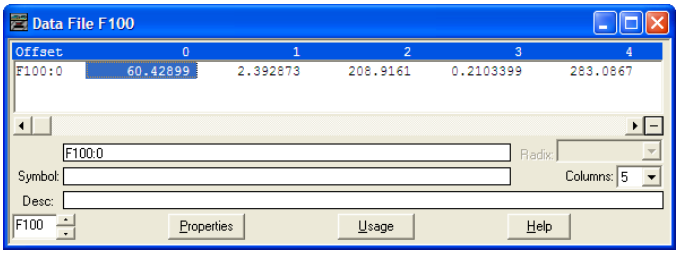
Figure 56 - Example Scattered Read Request Data File



MicroLogix 1100/1400 Controller Example Scattered Read Response Data

In this message example, we use the data table addresses in Figure 57 to store the response values that were read from the requested drive parameters. These values have been converted using a CPW (Copy Word) instruction for correct presentation.

Figure 57 - Example Scattered Read Response Data File



In this example, the parameters have the following values:

PowerFlex 750-Series Drive Parameter	Address	Read Value
1 - [Output Frequency]	F100:0	60.42899 Hz
7 - [Output Current]	F100:1	2.392873 Amp
8 - [Output Voltage]	F100:2	208.9161V AC
9 - [Output Power]	F100:3	0.2103399 kW
11 - [DC Bus Voltage]	F100:4	283.0867V DC

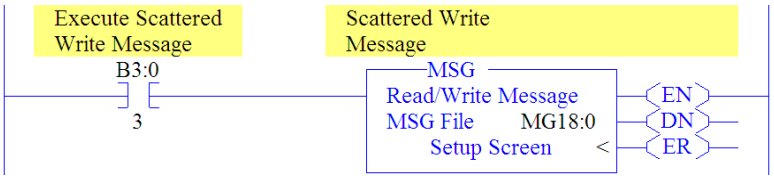
MicroLogix 1100/1400 Controller Example Ladder Logic Program to Write Multiple Parameters

A Custom scattered write message is used to write to multiple parameters. This write message example writes the following values to these five 32-bit REAL (floating point) parameters in a PowerFlex 750-Series drive:

PowerFlex 750-Series Drive Parameter	Write Value
536 - [Accel Time 2]	11.1 Sec
538 - [Decel Time 2]	22.2 Sec
575 - [Preset Speed 5]	33.3 Hz
576 - [Preset Speed 6]	44.4 Hz
577 - [Preset Speed 7]	55.5 Hz

See DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F) for parameter numbering.

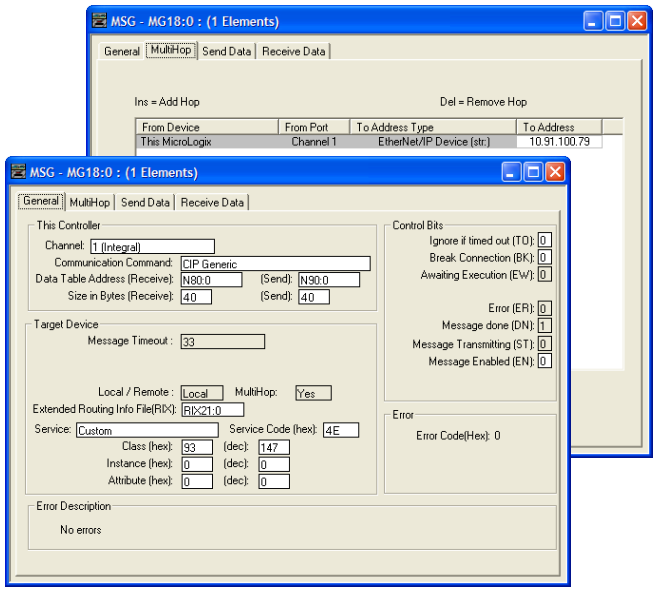
Figure 58 - Example Ladder Logic to Write Multiple Parameters



IMPORTANT If the explicit message scattered write must be written continuously, then use a separate Generic Set service explicit message single write for each parameter using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F and attribute A (10 decimal; see page 74). Attribute A writes to RAM—not NVS (EEPROM) memory. This example scattered write message using attribute 0 writes to NVS. Over time, continuous writes will exceed the EEPROM life cycle and cause the drive to malfunction.

MicroLogix 1100/1400 Controller – Formatting a Message to Write Multiple Parameters

Figure 59 - Custom Scattered Write Multiple Message Configuration Dialog Boxes



The following table identifies the data that is required in each box to configure a message to write multiple parameters.

General Tab	Example Value	Description
Channel	1	Controller port to which the network is connected.
Comm… Command	CIP Generic	Used to access the DPI Parameter Object in the option module.
Data Table Address		An unused controller data table address containing the message instruction.
Receive	N80:0	This address is the starting word of the response file.
Send	N90:0	This address is the starting word of the request file.
Size in Bytes		Each byte size is an 8-bit integer.
Receive	40 (2)	Number of bytes to be received.
Send	40 (2)	Number of bytes to be sent.
Extended Routing…	RIX21:0	An unused routing information file for the controller.
Service (1)	Custom	Required for scattered messages.
Service Code	4E (Hex.) (3)	Code for the requested service.
Class	93 or 9F (Hex.) (4)	Class ID for the DPI Parameter Object.
Instance	0 (Dec.)	Required for scattered messages.
Attribute	0 (Dec.)	Required for scattered messages.
MultiHop Tab	**Example Value**	**Description**
To Address	10.91.100.79	IP address of the option module connected to the drive.

(1) The default setting for Service is 'Custom', enabling entry of a Service Code not available from the Service pull-down menu. When choosing a Service other than 'Custom' from the pull-down menu, an appropriate Hex. value is automatically assigned to the Service Code box that is dimmed (unavailable).

(2) In this example, we are writing to five 32-bit REAL (floating point) parameters. Each parameter being written to requires four contiguous 16-bit words. Scattered write messages always assume that every parameter being written to is a 32-bit parameter, regardless of its actual size. Therefore, the Size in Bytes must be set to 40. The data structure format is shown on page 82. Maximum length is 128 words (256 bytes), which equates to 32 parameters. For parameter numbering, see DPI Parameter Object on page 94 (Class code 0x93) or Host DPI Parameter Object on page 108 (Class code 0x9F).

(3) Service Code 4E write messages are written to the drive's Nonvolatile Storage (EEPROM) memory, so the parameter value will remain even after the drive is power cycled. **Important:** Be very cautious as the EEPROM may quickly exceed its life cycle and cause the drive to malfunction.

(4) See Table 6 on page 50 for limitations of PowerFlex 750-Series drives when using DPI Parameter Object Class code 0x93 or Host DPI Parameter Object Class code 0x9F for explicit messaging.

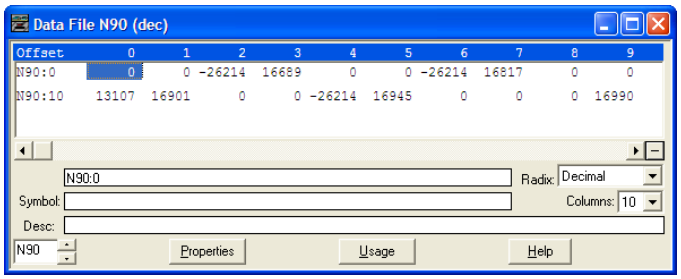
MicroLogix 1100/1400 Controller Example Scattered Write Request Data

In this message example, we use the F101: data table addresses to store the request values to be written to these 32-bit REAL (floating point) parameters:

PowerFlex 750-Series Drive Parameter	Address	Write Value
536 - [Accel Time 2]	F101:0	11.1 Sec
538 - [Decel Time 2]	F101:1	22.2 Sec
575 - [Preset Speed 5]	F101:2	33.3 Hz
576 - [Preset Speed 6]	F101:3	44.4 Hz
577 - [Preset Speed 7]	F101:4	55.5 Hz

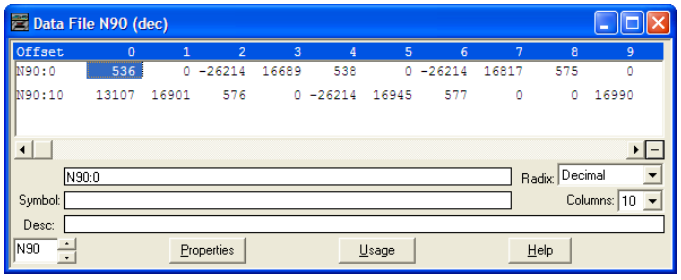
Figure 60 shows the parameter values, which, in this example, have been converted using a CPW (Copy Word) instruction—one for each value—to correctly write their values. The CPW instruction separates the 32-bit REAL (floating point) value (for example, F101:0 that contains 11.1 seconds) into two 16-bit integers (for example N90:2 and N90:3). While the values in the 16-bit integer registers represent the actual values being written in the 32-bit floating point registers, they will not appear correct, but the message will properly decode them.

Figure 60 - Example Scattered Write Request Unconverted Data File



To complete message configuration, the numbers of the parameters being written to must now be entered in the appropriate N90: data table registers as shown in Figure 61 for this example.

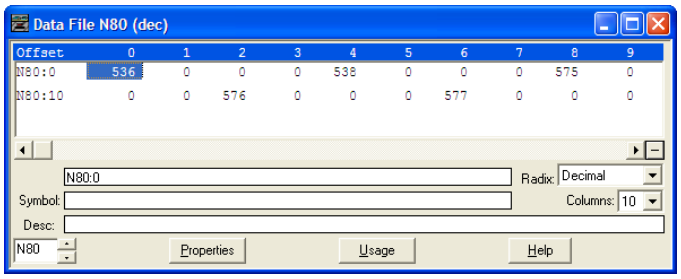
Figure 61 - Example Scattered Write Request Unconverted Data File with Entered Parameters



MicroLogix 1100/1400 Controller Example Scattered Write Response Data

In this message example, we use the data table addresses in Figure 62 to store the response values that were written to the requested drive parameters. Values of '0' indicate no errors occurred.

Figure 62 - Example Scattered Write Response Data File



TIP To verify that the parameter values were successfully written, use the HIM, DriveExplorer software, or DriveExecutive software to access the parameters and view their newly written values.

MicroLogix 1100/1400 Controller – Explanation of Request and Response Data for Read/Write Multiple Messaging

The data structures in Figure 63 use 32-bit words and can accommodate up to 32 parameters in a single message. In the Response Message, a parameter number with Bit 15 set indicates that the associated parameter value field contains an error code (parameter number in response data will be negative).

Figure 63 - Data Structures for Scattered Read/Write Messages

	Request (Source Data)		Response (Destination Data)
Word 0	Parameter Number (LSW)	Word 0	Parameter Number (LSW)
1	Parameter Number (MSW)	1	Parameter Number (MSW)
2	Parameter Value (LSW)	2	Parameter Value (LSW)
3	Parameter Value (MSW)	3	Parameter Value (MSW)
4	Parameter Number (LSW)	4	Parameter Number (LSW)
5	Parameter Number (MSW)	5	Parameter Number (MSW)
6	Parameter Value (LSW)	6	Parameter Value (LSW)
7	Parameter Value (MSW)	7	Parameter Value (MSW)
8	Parameter Number (LSW)	8	Parameter Number (LSW)
9	Parameter Number (MSW)	9	Parameter Number (MSW)
10	Parameter Value (LSW)	10	Parameter Value (LSW)
11	Parameter Value (MSW)	11	Parameter Value (MSW)
12	Parameter Number (LSW)	12	Parameter Number (LSW)
13	Parameter Number (MSW)	13	Parameter Number (MSW)
14	Parameter Value (LSW)	14	Parameter Value (LSW)
15	Parameter Value (MSW)	15	Parameter Value (MSW)
16	Parameter Number (LSW)	16	Parameter Number (LSW)
17	Parameter Number (MSW)	17	Parameter Number (MSW)
18	Parameter Value (LSW)	18	Parameter Value (LSW)
19	Parameter Value (MSW)	19	Parameter Value (MSW)
20	Parameter Number (LSW)	20	Parameter Number (LSW)
21	Parameter Number (MSW)	21	Parameter Number (MSW)
22	Parameter Value (LSW)	22	Parameter Value (LSW)
23	Parameter Value (MSW)	23	Parameter Value (MSW)
24	Parameter Number (LSW)	24	Parameter Number (LSW)
25	Parameter Number (MSW)	25	Parameter Number (MSW)
26	Parameter Value (LSW)	26	Parameter Value (LSW)
27	Parameter Value (MSW)	27	Parameter Value (MSW)
28	Parameter Number (LSW)	28	Parameter Number (LSW)
29	Parameter Number (MSW)	29	Parameter Number (MSW)
30	Parameter Value (LSW)	30	Parameter Value (LSW)
31	Parameter Value (MSW)	31	Parameter Value (MSW)
32	Parameter Number (LSW)	32	Parameter Number (LSW)
33	Parameter Number (MSW)	33	Parameter Number (MSW)
34	Parameter Value (LSW)	34	Parameter Value (LSW)
35	Parameter Value (MSW)	35	Parameter Value (MSW)
⋮		⋮	
124	Parameter Number (LSW)	124	Parameter Number (LSW)
125	Parameter Number (MSW)	125	Parameter Number (MSW)
126	Parameter Value (LSW)	126	Parameter Value (LSW)
127	Parameter Value (MSW)	127	Parameter Value (MSW)

EtherNet/IP Objects

This appendix presents information about the EtherNet/IP objects that can be accessed using Explicit Messages. For information on the format of Explicit Messages and example ladder logic programs, see Chapter 3, Using Explicit Messaging.

TIP See the EtherNet/IP specification for more information about EtherNet/IP objects. Information about the EtherNet/IP specification is available on the ODVA website (http://www.odva.org).

Supported Data Types

Data Type	Description
BOOL	8-bit value -- low bit is true or false
BOOL[x]	Array of n bits
CONTAINER	32-bit parameter value - sign extended if necessary
DINT	32-bit signed integer
INT	16-bit signed integer
LWORD	64-bit unsigned integer
REAL	32-bit floating point
SHORT_STRING	Struct of: USINT length indicator (L); USINT[L] characters
SINT	8-bit signed integer
STRINGN	Struct of: UINT character length indicator (W); UINT length indicator (L); USINT[W x L] string data
STRING[x]	Array of n characters
STRUCT	Structure name only - no size in addition to elements
TCHAR	8 or 16-bit character
UDINT	32-bit unsigned integer
UINT	16-bit unsigned integer
USINT	8-bit unsigned integer

Identity Object

Class Code

Hexadecimal	Decimal
0x01	1

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x05	No	Yes	Reset
0x0E	Yes	Yes	Get_Attribute_Single
0x01	Yes	Yes	Get_Attributes_All

Instances

The number of instances depends on the number of components in the device connected to the option module. This number of components can be read in Instance 0, Attribute 2.

Instance	Description
0	Class
1	Host
2…15	Peripherals on Ports 1…14

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
2	Get	Max Instance	UINT	Total number of instances

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Vendor ID	UINT	1 = Allen-Bradley
2	Get	Device Type	UINT	142
3	Get	Product Code	UINT	Number identifying product name and rating
4	Get	Revision: Major Minor	STRUCT of: USINT USINT	 Value varies Value varies
5	Get	Status	UINT	Bit 0 = Owned Bit 8 = Minor recoverable fault Bit 10 = Major recoverable fault
6	Get	Serial Number	UDINT	Unique 32-bit number
7	Get	Product Name	SHORT_STRING	Product name and rating

Assembly Object

Class Code

Hexadecimal	Decimal
0x04	4

Services

	Implemented for:		
Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

Instance	Description
1	All I/O data being read from the DPI device (read-only)
2	All I/O data written to the DPI device (read/write)

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	2
2	Get	Max Instance	UINT	2
100	Set	Control Timeout	UINT	Control timeout in seconds

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Number of Members	UINT	1
2	Get	Member List	ARRAY of STRUCT: UINT UINT Packed EPATH	 Size of member data Size of member path Member path
3	Conditional (1)	Data	Array of Bits	Data to be transferred
4	Get	Size	UINT	Size of assembly data in bits

(1) For instance 1, access rule for the data attribute is Get. For instance 2, it is Get/Set.

IMPORTANT Setting an assembly object attribute can be done only when the Control Timeout (class attribute 100) has been set to a non-zero value.

Register Object

Class Code

Hexadecimal	Decimal
0x07	7

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

Instance	Description
1	All I/O data being read from the option module (read-only)
2	All I/O data written to the option module (read/write)
3	Logic Status and Feedback data (read-only)
4	Logic Command and Reference data (read/write)
5	DL To Net 01 (input data from option module to scanner) (read only)
6	DL From Net 01 (output data from scanner to option module) (read/write)
⋮	⋮
35	DL To Net 16 (input data from option module to scanner) (read only)
36	DL From Net 16 (output data from scanner to option module) (read/write)
37	Logic Status and Feedback data (read-only)
38	Masked Logic Command (1) (read/write)
39	Logic Status data (read-only)
40	Logic Command data (read/write)
41	Feedback data (read-only)
42	Reference data (read/write)

(1) The mask command DWORD is set to the value of the first DWORD of the data where there are ones in the second DWORD of the data. Only the bits of the Logic Command that have the corresponding mask bit set are applied.

Class Attributes

Attribute ID	Access Rule	Description
1	Read	Revision
2	Read	Maximum Instance
3	Read	Number of Instances
100	Read/Write	Timeout

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Bad Flag	BOOL	If set to 1, then attribute 4 may contain invalid data. 0 = good 1 = bad
2	Get	Direction	BOOL	Direction of data transfer 0 = Producer Register (drive to network) 1 = Consumer Register (network to drive)
3	Get	Size	UINT	Size of register data in bits
4	Conditional (1)	Data	ARRAY of BITS	Data to be transferred

(1) For this attribute, the Access Rule is Get if Direction = 0. The Access Rule is Set if Direction = 1.

PCCC Object

Class Code

Hexadecimal	Decimal
0x67	103

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x4B	No	Yes	Execute_PCCC
0x4C	No	Yes	Execute_DH+

Instances

Supports Instance 1.

Class Attributes

Not supported.

Instance Attributes

Not supported.

Message Structure for Execute_PCCC

Request		
Name	**Data Type**	**Description**
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor
Other	Product Specific	Identifier of user, task, etc. on the requestor
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code. Not used for all CMDs.
PCCC_params	ARRAY of USINT	CMD/FNC specific parameters

Response		
Name	**Data Type**	**Description**
Length	USINT	Length of requestor ID
Vendor	UINT	Vendor number of requestor
Serial Number	UDINT	ASA serial number of requestor
Other	Product Specific	Identifier of user, task, etc. on the requestor
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word. Same value as the request.
EXT_STS	USINT	Extended status. Not used for all CMDs.
PCCC_results	ARRAY of USINT	CMD/FNC specific result data

Message Structure for Execute_DH+

Request		
Name	**Data Type**	**Description**
DLink	UINT	Destination Link ID
DSta	USINT	Destination Station number
DUser	USINT	Destination 'User' number
SLink	UINT	Source Link ID
SSta	USINT	Source Station number
SUser	USINT	Source User number
CMD	USINT	Command byte
STS	USINT	0
TNSW	UINT	Transport word
FNC	USINT	Function code; not used for all CMDs
PCCC_params	ARRAY of USINT	CMD/FNC specific parameters

Response		
Name	**Data Type**	**Description**
DLink	UINT	Destination Link ID
DSta	USINT	Destination Station number
DUser	USINT	Destination 'User' number
SLink	UINT	Source Link ID
SSta	USINT	Source Station number
SUser	USINT	Source User number
CMD	USINT	Command byte
STS	USINT	Status byte
TNSW	UINT	Transport word. Same value as the request.
EXT_STS	USINT	Extended Status; not used for all CMDs
PCCC_results	ARRAY of USINT	CMD/FNC specific result data

The EtherNet/IP option module supports the following PCCC command types:

CMD	FNC	Description
0x06	0x03	Identify host and some status
0x0F	0x67	PLC-5 typed write
0x0F	0x68	PLC-5 typed read
0x0F	0x95	Encapsulate other protocol
0x0F	0xA2	SLC 500 protected typed read with 3 address fields
0x0F	0xAA	SLC 500 protected typed write with 3 address fields
0x0F	0xA1	SLC 500 protected typed read with 2 address fields
0x0F	0xA9	SLC 500 protected typed write with 2 address fields
0x0F	0x00	Word range read
0x0F	0x01	Word range write

For more information regarding PCCC commands, see DF1 Protocol and Command Set Reference Manual, publication 1770-6.5.16.

N-Files

N-File	Description	
N42	This N-file lets you read and write some values configuring the port.	
N42:3	Time-out (read/write): Time (in seconds) allowed between messages to the N45 file. If the option module does not receive a message in the specified time, it performs the fault action configured in its [Comm Flt Action] parameter. A valid setting is between 1 and 32767 seconds (5 . . . 20 seconds is recommended).	
N42:7	Option module Port Number (read only): Drive port in which option module resides.	
N42:8	Peer Option Modules (read only): Bit field of devices with peer messaging capabilities.	
N45	This N-file lets you read and write control I/O messages. You can write control I/O messages only when all of the following conditions are true: • The option module is not receiving I/O from a scanner. For example, there is no scanner on the network, the scanner is in idle (program) mode, the scanner is faulted, or the option module is not mapped to the scanner. • The option module is not receiving Peer I/O from another option module. • The value of N42:3 is set to a non-zero value.	
	Write	**Read**
N45:0	Logic Command (least significant)	Logic Status (least significant)
N45:1	Logic Command (most significant)	Logic Status (most significant)
N45:2	Reference (least significant)	Feedback (least significant)
N45:3	Reference (most significant)	Feedback (most significant)
N45:4	DL From Net 01 (least significant)	DL To Net 01 (least significant)
N45:5	DL From Net 01 (most significant)	DL To Net 01 (most significant)
N45:6	DL From Net 02 (least significant)	DL To Net 02 (least significant)
N45:7	DL From Net 02 (most significant)	DL To Net 02 (most significant)
N45:8	DL From Net 03 (least significant)	DL To Net 03 (least significant)
N45:9	DL From Net 03 (most significant)	DL To Net 03 (most significant)
N45:10	DL From Net 04 (least significant)	DL To Net 04 (least significant)
N45:11	DL From Net 04 (most significant)	DL To Net 04 (most significant)
N45:12	DL From Net 05 (least significant)	DL To Net 05 (least significant)
N45:13	DL From Net 05 (most significant)	DL To Net 05 (most significant)
N45:14	DL From Net 06 (least significant)	DL To Net 06 (least significant)
N45:15	DL From Net 06 (most significant)	DL To Net 06 (most significant)
N45:16	DL From Net 07 (least significant)	DL To Net 07 (least significant)
N45:17	DL From Net 07 (most significant)	DL To Net 07 (most significant)
N45:18	DL From Net 08 (least significant)	DL To Net 08 (least significant)
N45:19	DL From Net 08 (most significant)	DL To Net 08 (most significant)
N45:20	DL From Net 09 (least significant)	DL To Net 09 (least significant)
N45:21	DL From Net 09 (most significant)	DL To Net 09 (most significant)
N45:22	DL From Net 10 (least significant)	DL To Net 10 (least significant)
N45:23	DL From Net 10 (most significant)	DL To Net 10 (most significant)
N45:24	DL From Net 11 (least significant)	DL To Net 11 (least significant)
N45:25	DL From Net 11 (most significant)	DL To Net 11 (most significant)
N45:26	DL From Net 12 (least significant)	DL To Net 12 (least significant)
N45:27	DL From Net 12 (most significant)	DL To Net 12 (most significant)
N45:28	DL From Net 13 (least significant)	DL To Net 13 (least significant)
N45:29	DL From Net 13 (most significant)	DL To Net 13 (most significant)
N45:30	DL From Net 14 (least significant)	DL To Net 14 (least significant)
N45:31	DL From Net 14 (most significant)	DL To Net 14 (most significant)
N45:32	DL From Net 15 (least significant)	DL To Net 15 (least significant)
N45:33	DL From Net 15 (most significant)	DL To Net 15 (most significant)
N45:34	DL From Net 16 (least significant)	DL To Net 16 (least significant)
N45:35	DL From Net 16 (most significant)	DL To Net 16 (most significant)

N-File	Description
N150…N199	These N-files let you read and write parameter values in the PowerFlex 750-Series drive as 32-bit double words. You can interpret the data in various ways (for example, 32-bit real, 32-bit integer) To read a parameter, you need to send a message with two elements. For example, to read parameter 1, read two elements beginning at N150:2. As another example, to read parameters 2…6, read ten elements beginning at N150:4.
N150:0…1	Number of parameters in the drive
N150:2…249	Drive parameters 1…124
N151:0…249	Drive parameters 125…249
N152:0…249	Drive parameters 250…374
N153:0…249	Drive parameters 375…499
⋮	⋮
N199:0…249	Drive parameters 6125…6249
N201…N212	These N-files let you read and write parameter values in DPI Peripherals (for example, a HIM or option module) as 32-bit double words. You can interpret the data in various ways (for example, 32-bit real, 32-bit integer) To read a parameter, you need to send a message with two elements. For example, to read parameter 1 in the peripheral connected to DPI port 1, read two elements beginning at N201:2. As another example, to read parameters 2…6 in the peripheral connected to DPI port 5 (the option module), read ten elements beginning at N209:4.
N201:0…1	Number of parameters in the DPI peripheral at DPI port 1
N201:2…249	Parameters 1…124 in the DPI peripheral at DPI port 1
N202:0…249	Parameters 125…249 in the DPI peripheral at DPI port 1
N203:0…1	Number of parameters in the DPI peripheral at DPI port 2
N203:2…249	Parameters 1…124 in the DPI peripheral at DPI port 2
N204:0…249	Parameters 125…249 in the DPI peripheral at DPI port 2
N205:0…1	Number of parameters in the DPI peripheral at DPI port 3
N205:2…249	Parameters 1…124 in the DPI peripheral at DPI port 3
N206:0…249	Parameters 125…249 in the DPI peripheral at DPI port 3
N207:0…1	Number of parameters in the DPI peripheral at DPI port 4
N207:2…249	Parameters 1…124 in the DPI peripheral at DPI port 4
N208:0…249	Parameters 125…249 in the DPI peripheral at DPI port 4
N209:0…1	Number of parameters in the DPI peripheral at DPI port 5
N209:2…249	Parameters 1…124 in the DPI peripheral at DPI port 5
N210:0…249	Parameters 125…249 in the DPI peripheral at DPI port 5
N211:0…1	Number of parameters in the DPI peripheral at DPI port 6
N211:2…249	Parameters 1…124 in the DPI peripheral at DPI port 6
N212:0…249	Parameters 125…249 in the DPI peripheral at DPI port 6
N213:0…1	Number of parameters in the DPI peripheral at DPI port 7
N213:2…249	Parameters 1…124 in the DPI peripheral at DPI port 7
N214:0…249	Parameters 125…249 in the DPI peripheral at DPI port 7
N215:0…1	Number of parameters in the DPI peripheral at DPI port 8
N215:2…249	Parameters 1…124 in the DPI peripheral at DPI port 8
N216:0…249	Parameters 125…249 in the DPI peripheral at DPI port 8
N217:0…1	Number of parameters in the DPI peripheral at DPI port 9
N217:2…249	Parameters 1…124 in the DPI peripheral at DPI port 9
N218:0…249	Parameters 125…249 in the DPI peripheral at DPI port 9
N219:0…1	Number of parameters in the DPI peripheral at DPI port 10
N219:2…249	Parameters 1…124 in the DPI peripheral at DPI port 10
N220:0…249	Parameters 125…249 in the DPI peripheral at DPI port 10
N221:0…1	Number of parameters in the DPI peripheral at DPI port 11
N221:2…249	Parameters 1…124 in the DPI peripheral at DPI port 11
N222:0…249	Parameters 125…249 in the DPI peripheral at DPI port 11
N223:0…1	Number of parameters in the DPI peripheral at DPI port 12
N223:2…249	Parameters 1…124 in the DPI peripheral at DPI port 12
N224:0…249	Parameters 125…249 in the DPI peripheral at DPI port 12
N225:0…1	Number of parameters in the DPI peripheral at DPI port 13
N225:2…249	Parameters 1…124 in the DPI peripheral at DPI port 13
N226:0…249	Parameters 125…249 in the DPI peripheral at DPI port 13
N227:0…1	Number of parameters in the DPI peripheral at DPI port 14
N227:2…249	Parameters 1…124 in the DPI peripheral at DPI port 14
N228:0…249	Parameters 125…249 in the DPI peripheral at DPI port 14

DPI Device Object

Class Code

Hexadecimal	Decimal
0x92	146

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the number of components in the device. The total number of components can be read in Instance 0, Class Attribute 4.

Instances		Device
(Hex.)	(Dec.)	
0x0000…0x3FFF	0…16383	Host Drive
0x4000…0x43FF	16384…17407	Option Module
0x4400…0x47FF	17408…18431	Port 1
0x4800…0x4BFF	18432…19455	Port 2
0x4C00…0x4FFF	19456…20479	Port 3
0x5000…0x53FF	20480…21503	Port 4
0x5400…0x57FF	21504…22527	Port 5
0x5800…0x5BFF	22528…23551	Port 6
0x5C00…0x5FFF	23552…24575	Port 7
0x6000…0x63FF	24576…25599	Port 8
0x6400…0x67FF	25600… 26623	Port 9
0x6800…0x6BFF	26624…27647	Port 10
0x6C00…0x6FFF	27648…28671	Port 11
0x7000…0x73FF	28672…29695	Port 12
0x7400…0x77FF	29696…30719	Port 13
0x7800…0x7BFF	30720…31743	Port 14

Example	Description
0	Class Attributes (Drive)
1	Drive Component 1
2	Drive Component 2
⋮	⋮
16384	Class Attributes (Option Module)
16385	Option Module Component 1
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Family Code	USINT	0x00 = DPI Peripheral 0x90 = PowerFlex 750-Series Drive 0xA0 = 20-750-xxxx Option Module 0xFF = HIM
1	Get	Family Text	STRING[16]	Text identifying the device.
2	Set	Language Code	USINT	0 = English 1 = French 2 = Spanish 3 = Italian 4 = German 5 = Japanese 6 = Portuguese 7 = Mandarin Chinese 8 = Russian 9 = Dutch 10 = Korean
3	Get	Product Series	USINT	1 = A 2 = B …
4	Get	Number of Components	USINT	Number of components (for example, main control board, I/O boards) in the device.
5	Set	User Definable Text	STRING[16]	Text identifying the device with a user-supplied name.
6	Get	Status Text	STRING[12]	Text describing the status of the device.
7	Get	Configuration Code	USINT	Identification of variations.
8	Get	Configuration Text	STRING[16]	Text identifying a variation of a family device.
9	Get	Brand Code	UINT	0x0001 = Allen-Bradley
11	Get	NVS Checksum	UINT	Checksum of the Nonvolatile Storage in a device.
12	Get	Class Revision	UINT	2 = DPI
13	Get	Character Set Code	USINT	0 = SCANport HIM 1 = ISO 8859-1 (Latin 1) 2 = ISO 8859-2 (Latin 2) 3 = ISO 8859-3 (Latin 3) 4 = ISO 8859-4 (Latin 4) 5 = ISO 8859-5 (Cyrillic) 6 = ISO 8859-6 (Arabic) 7 = ISO 8859-7 (Greek) 8 = ISO 8859-8 (Hebrew) 9 = ISO 8859-9 (Turkish) 10 = ISO 8859-10 (Nordic) 255 = ISO 10646 (Unicode)
14	Get	Product Option Support	BOOL[64]	
15	Get	Languages Supported	STRUCT of: USINT USINT[n]	 Number of Languages Language Codes (see Class Attribute 2)
16	Get	Date of Manufacture	STRUCT of: UINT USINT USINT	 Year Month Day
17	Get	Product Revision	STRUCT of: USINT USINT	 Major Firmware Release Minor Firmware Release
18	Get	Serial Number	UDINT	Value between 0x00000000 and 0xFFFFFFFF
19	Set	Language Selected	USINT	0 = Default (HIM will prompt at startup) 1 = Language was selected (no prompt)

Attribute ID	Access Rule	Name	Data Type	Description
20	Set	Customer-Generated Firmware	STRING[36]	GUID (Globally Unique Identifier) identifying customer firmware upgraded into the device.
30	Get	International Status Text	STRINGN	Text describing the status of device with support for Unicode.
31	Get/Set	International User Definable Text	STRINGN	Text identifying the device with a user-supplied name with support for Unicode.
34	Get	Key Information	STRUCT of: UDINT UDINT UINT UINT UINT USINT USINT USINT USINT USINT USINT[16]	 Rating Code Device Serial Number Customization Code Customization Revision Brand Code Family Code Config Code Language Code Major Revision Minor Revision Customer-Generated Firmware UUID
35	Get	NVS CRC	UDINT	A 32-bit CRC of the Nonvolatile Storage in a device.
38	Set	ADC Configuration Signature	USINT[16]	Value stored by the device and zeroed if its configuration changes.
39	Get	SI Driver Code	UINT	Code identifying the protocol between the device and host.
128	Get	Customization Code	UINT	Code identifying the customized device.
129	Get	Customization Revision Number	UINT	Revision of the customized device.
130	Get	Customization Device Text	STRING[32]	Text identifying the customized device.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
3	Get	Component Name	STRING[32]	Name of the component
4	Get	Component Firmware Revision	STRUCT of: USINT USINT	 Major Revision Minor Revision
8	Get	Component Serial Number	UDINT	Value between 0x00000000 and 0xFFFFFFFF
9	Get	International Component Name	STRINGN	Name of the component with support for Unicode.

DPI Parameter Object

Class Code

Hexadecimal	Decimal
0x93	147

To access 'Host Config' parameters, use the Host DPI Parameter Object (Class Code 0x9F).

Instances

The number of instances depends on the number of parameters in the device. The total number of parameters can be read in Instance 0, Attribute 0.

Instances		Device
(Hex.)	(Dec.)	
0x0000…0x3FFF	0…16383	Host Drive
0x4000…0x43FF	16384…17407	Option Module
0x4400…0x47FF	17408…18431	Port 1
0x4800…0x4BFF	18432…19455	Port 2
0x4C00…0x4FFF	19456…20479	Port 3
0x5000…0x53FF	20480…21503	Port 4
0x5400…0x57FF	21504…22527	Port 5
0x5800…0x5BFF	22528…23551	Port 6
0x5C00…0x5FFF	23552…24575	Port 7
0x6000…0x63FF	24576…25599	Port 8
0x6400…0x67FF	25600…26623	Port 9
0x6800…0x6BFF	26624…27647	Port 10
0x6C00…0x6FFF	27648…28671	Port 11
0x7000…0x73FF	28672…29695	Port 12
0x7400…0x77FF	29696…30719	Port 13
0x7800…0x7BFF	30720…31743	Port 14

Example	Description
0	Class Attributes (Drive)
1	Drive Parameter 1 Attributes
2	Drive Parameter 2 Attributes
⋮	⋮
16384	Class Attributes (Option Module)
16385	Option Module Parameter 1 Attributes
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Number of Instances	UINT	Number of parameters in the device
1	Set	Write Protect Password	UINT	0 = Password disabled n = Password value
2	Set	NVS Command Write	USINT	0 = No Operation 1 = Store values in active memory to NVS 2 = Load values in NVS to active memory 3 = Load default values to active memory 4 = Partial defaults 5 = System defaults
3	Get	NVS Parameter Value Checksum	UINT	Checksum of all parameter values in a user set in NVS
4	Get	NVS Link Value Checksum	UINT	Checksum of parameter links in a user set in NVS
5	Get	First Accessible Parameter	UINT	First parameter available if parameters are protected by passwords. A '0' indicates all parameters are protected.
7	Get	Class Revision	UINT	2 = DPI
8	Get	First Parameter Processing Error	UINT	The first parameter that has been written with a value outside of its range. A '0' indicates no errors.
9	Set	Link Command	USINT	0 = No Operation 1 = Clear All Parameter Links (This does not clear links to function blocks.)

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
6	Get	DPI Offline Read Full	STRUCT of:	
			BOOL[32]	Descriptor
			CONTAINER	Offline Minimum value
			CONTAINER	Offline Maximum value
			CONTAINER	Offline Default value
			STRING[16]	Parameter name
			STRING[4]	Offline parameter units
			UINT	Online minimum parameter instance
			UINT	Online maximum parameter instance
			UINT	Online default parameter instance
			UINT	Multiplier parameter instance
			UINT	Divisor parameter instance
			UINT	Base parameter instance
			UINT	Offset parameter instance
			USINT	Formula number
			USINT	Pad byte (always zero)
			UINT	Help instance
			UINT	Pad word (always a value of zero)
			CONTAINER	Parameter value
			UINT	Multiplier
			UNIT	Divisor
			UNIT	Base
			INT	Offset
7	Get	DPI Online Read Full	STRUCT of:	
			BOOL[32]	Descriptor (see page 97)
			CONTAINER[1]	Parameter value
			CONTAINER	Minimum value
			CONTAINER	Maximum value
			CONTAINER	Default value
			UINT	Next parameter
			UINT	Previous parameter
			STRING[4]	Units (for example, Amps, Hz)
			UINT	Multiplier [2]
			UINT	Divisor [2]
			UINT	Base [2]
			INT	Offset [2]
			USINT[3]	Link (source of the value) (0 = no link)
			USINT	Always zero (0)
			STRING[16]	Parameter name
8	Get	DPI Descriptor	BOOL[32]	Descriptor (see page 97)
9	Get/Set	DPI Parameter Value	Various	Parameter value in NVS. [3]
10	Get/Set	DPI RAM Parameter Value	Various	Parameter value in temporary memory. Valid only for DPI drives.
11	Get/Set	DPI Link	USINT[3]	Link (parameter or function block that is the source of the value) (0 = no link)
12	Get	Help Object Instance	UINT	ID for help text for this parameter
13	Get	DPI Read Basic	STRUCT of:	
			BOOL[32]	Descriptor (see page 97)
			CONTAINER	Parameter value
			CONTAINER	Minimum value
			CONTAINER	Maximum value
			CONTAINER	Default value
			STRING[16]	Parameter name
			STRING[4]	Units (for example, Amps, Hz)
14	Get	DPI Parameter Name	STRING[16]	Parameter name
15	Get	DPI Parameter Alias	STRING[16]	Customer-supplied parameter name.

Attribute ID	Access Rule	Name	Data Type	Description
16	Get	Parameter Processing Error	USINT	0 = No error 1 = Value is less than the minimum 2 = Value is greater than the maximum
18	Get	International DPI Offline Parameter Text	Struct of: STRINGN STRINGN	 International parameter name International offline units
19	Get	International DPI Online Parameter Text	Struct of: STRINGN STRINGN	 International parameter name International online units
20	Get	International DPI Online Read Full	Struct of: BOOL[32] CONTAINER CONTAINER CONTAINER CONTAINER UINT UINT UINT UINT UINT INT USINT[3] USINT BOOL[32] STRINGN STRINGN	 Descriptor Parameter value Online minimum value Online maximum value Online default value Next Previous Multiplier Divisor Base Offset Link Pad word (always zero) Extended descriptor International parameter name International online parameter units
21	Get	DPI Extended Descriptor	UDINT	Extended Descriptor (see page 98)
22	Get	International DPI Offline Read Full	Struct of: BOOL CONTAINER CONTAINER CONTAINER UINT UINT UINT UINT UINT UINT UINT USINT USINT UINT UINT CONTAINER UINT UINT UINT INT BOOL[32] STRINGN STRINGN	 Descriptor Offline minimum value Offline maximum value Offline default value Online minimum parameter instance Online maximum parameter instance Online default parameter instance Multiplier parameter instance Divisor parameter instance Base parameter instance Offset parameter instance Formula number Pad word (always zero) Help instance Pad word (always a value of zero) Parameter value Multiplier Divisor Base Offset Extended DPI descriptor International DPI parameter name International DPI offline parameter units

(1) A CONTAINER is a 32-bit block of data that contains the data type used by a parameter value. If signed, the value is sign extended. Padding is used in the CONTAINER to be sure that it is always 32-bits.

(2) This value is used in the formulas used to convert the parameter value between display units and internal units. See Formulas for Converting on page 99.

(3) Do NOT continually write parameter data to NVS. See the attention on page 49.

Descriptor Attributes

Bit	Name	Description
0	Data Type (Bit 1)	Right bit is least significant bit (0). 000 = USINT used as an array of Boolean 001 = UINT used as an array of Boolean 010 = USINT (8-bit integer) 011 = UINT (16-bit integer) 100 = UDINT (32-bit integer) 101 = TCHAR ((8-bit (not Unicode) or 16-bits (Unicode)) 110 = REAL (32-bit floating point value) 111 = Use bits 16, 17, 18
1	Data Type (Bit 2)	
2	Data Type (Bit 3)	
3	Sign Type	0 = unsigned 1 = signed
4	Hidden	0 = visible 1 = hidden
5	Not a Link Sink	0 = May be the sink end of a link 1 = May not be the sink end of a link
6	Not Recallable	0 = Recallable from NVS 1 = Not Recallable from NVS
7	ENUM	0 = No ENUM text 1 = ENUM text
8	Writable	0 = Read only 1 = Read/write
9	Not Writable When Enabled	0 = Writable when enabled (for example, drive running) 1 = Not writable when enabled
10	Instance	0 = Parameter value is not a Reference to another parameter 1 = Parameter value refers to another parameter
11	Uses Bit ENUM Mask	This parameter instance supports the Bit ENUM Mask attribute. For more information, see the definition of the attribute.
12	Decimal Place (Bit 0)	Number of digits to the right of the decimal point. 0000 = 0 1111 = 15
13	Decimal Place (Bit 1)	
14	Decimal Place (Bit 2)	
15	Decimal Place (Bit 3)	
16	Extended Data Type (Bit 4)	Bit 16 is the least significant bit. 000 = Reserved 001 = UDINT used as an array of Boolean 010 = Reserved 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved
17	Extended Data Type (Bit 5)	
18	Extended Data Type (Bit 6)	
19	Parameter Exists	Used to mark parameters that are not available to network tools.
20	Not Used	Reserved
21	Formula Links	Indicates the Formula Data is derived from other parameters.
22	Access Level (Bit 1)	A 3-bit field used to control access to parameter data.
23	Access Level (Bit 2)	
24	Access Level (Bit 3)	
25	Writable ENUM	ENUM text: 0 = Read Only, 1 = Read/Write
26	Not a Link Source	0 = May be the source end of a link 1 = May not be the source end of a link
27	Enhanced Bit ENUM	Parameter supports enhanced bit ENUMs.
28	Enhanced ENUM	Parameter supports enhanced ENUMs.
29	Uses DPI Limits Object	Parameter uses the DPI Limits Object. Intelligent offline tools make use of the Limits Object to select limits and units.
30	Extended Descriptor	Parameter uses Extended Descriptor bits, which can be obtained by reading the DPI Extended Descriptor attribute for this parameter.
31	Always Upload/Download	Parameter shall always be included in uploads and downloads.

Extended Descriptor Attributes

Bit	Name	Description
0	Indirect Mode	0 = Analog (selects entire parameters) 1 = Digital (selects individual bits within parameters)
1	Indirect Type 0	Analog input list (Instance 0xFFFF)
2	Indirect Type 1	Digital input list (Instance 0xFFFE)
3	Indirect Type 2	Feedback list (Instance 0xFFFD)
4	Indirect Type 3	Analog output list (Instance 0xFFFC)
5	Indirect Type 4	Digital output list (Instance 0xFFFB)
6	Indirect Type 5	Undefined (Instance 0xFFFA)
7	Indirect Type 6	Undefined (Instance 0xFFF9)
8	Indirect Type 7	Undefined (Instance 0xFFF8)
9	Indirect Type 8	Undefined (Instance 0xFFF7)
10	Indirect Type 9	Undefined (Instance 0xFFF6)
11	Indirect Type 10	Undefined (Instance 0xFFF5)
12	Indirect Type 11	Undefined (Instance 0xFFF4)
13	Indirect Type 12	Undefined (Instance 0xFFF3)
14	Indirect Type 13	Undefined (Instance 0xFFF2)
15	Indirect Type 14	Parameter-specific list
16	FP Max Decimals Bit 0	These four bits are used on REAL parameters only. They indicate the maximum number of decimal places to be displayed for small values. A value of 0 indicates to not limit the number of decimal places used.
17	FP Max Decimals Bit 1	
18	FP Max Decimals Bit 2	
19	FP Max Decimals Bit 1	
20	Extended Parameter Reference	0 = Not an Extended Parameter Reference 1 = Extended Parameter Reference An Extended Parameter Reference contains a reference to another parameter. The value is formatted the same as an analog mode Indirect Selector parameter (SSpppp, where SS = slot number of device to which this Extended Parameter Reference is pointing, and pppp = number of the parameter or diagnostic item to which this Extended Parameter Reference is pointing). Note that an Extended Parameter Reference can only select parameters unlike an Indirect Selector. An Extended Parameter Reference could be used to configure a Datalink or show the source of a Reference (among other uses).
21	Uses Rating Table Object	This parameter has rating-dependent defaults and limits that can be obtained from the Rating Table Object. The Offline Read Full will include the default value for the smallest rating and limits that will accommodate the full range of values allowed in the family of devices using this particular combination of Family Code and Config Code. The Online Read Full will include the rating-dependent default and limit values for this particular combination of Family Code, Config Code, and Rating Code.
22	Writable Referenced Parameter	This bit must be zero unless the parameter is an Extended Parameter Reference. If the parameter is an Extended Parameter Reference, then: 0 = The referenced parameter may be read-only or writable. 1 = The referenced parameter must always be writable (including while running).
23	Disallow Zero	This bit must be zero unless the parameter is an Indirect Selector or Extended Parameter Reference. If the parameter is an Indirect Selector or Extended Parameter Reference, then: 0 = Allow zero 1 = Disallow zero If this bit is cleared (indicating that a value of zero is allowed), the device must support the 'Zero Text' parameter attribute so that a software tool or HIM can obtain text from the Zero Text parameter attribute. If this bit is set (indicating that a value of zero is disallowed), a software tool or HIM will not allow the user to enter a value of zero.
24	Datalink Out	This bit is used by offline tools and indicates that this is a Datalink Out parameter. Bit 20 must also be set.
25	Datalink In	This bit is used by offline tools and indicates that this is a Datalink In parameter. Bits 20 and 22 must also be set.
26	Not Writable While IO Active	This parameter cannot be written if the I/O data being exchanged between the Host and the peripheral is valid.
27	Command Parameter	This parameter commands the drive to take an action, such as 'Reset Defaults' or 'Autotune', and then returns to a value of zero. Offline software tools will not allow setting this parameter to anything other than a value of zero. If an offline file contains a Command Parameter with a non-zero value, the offline software tool will change the value to zero. Note that command parameters cannot have values that do not return to zero.

Bit	Name	Description
28	Current Value Is Default	This bit identifies a parameter that will not change if a 'Reset Defaults' is commanded. For example, if a drive contains a Language parameter that is set to German, setting defaults will leave the parameter set to German. Likewise, if the parameter is set to French, setting defaults will leave the parameter set to French.
29	Use Zero Text	If the 'Disallow Zero' bit is set, this bit must be cleared. If the 'Disallow Zero' bit is cleared, then: 0 = Use Disabled Text parameter class attribute. 1 = Use Zero Text parameter instance attribute.
30-31	Reserved	Reserved

Formulas for Converting

Display Value = ((Internal Value + Offset) x Multiplier x Base) / (Divisor x $10^{\text{Decimal Places}}$))

Internal Value = ((Display Value x Divisor x $10^{\text{Decimal Places}}$) / (Multiplier x Base)) - Offset

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Object Specific Services

Service Code	Implemented for:		Service Name	Allocation Size (in bytes)	
	Class	Instance		Par. Number	Par. Value
0x4D	Yes	No	Get_Attributes_Scattered	4	4
0x4E	Yes	No	Set_Attributes_Scattered	4	4

The table below lists the parameters for the Get_Attributes_Scattered and Set_Attributes_Scattered object-specific service:

Name	Data Type	Description
Parameter Number	UDINT	Parameter to read or write
Parameter Value	UDINT	Parameter value write (zero when reading)

The response data appears in the following format:

Name	Data Type	Description
Parameter Number	UDINT	Parameter read or write (1)
Parameter Value	UDINT	Parameter value read (zero when writing) (2)

(1) If an error occurred, bit 15 will be turned on in the response.

(2) If an error occurred, the error code will appear instead of the value.

DPI Fault Object

Class Code

Hexadecimal	Decimal
0x97	151

Products such as PowerFlex drives use this object for faults. Option Modules use this object for events.

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the maximum number of faults or events supported in the queue. The maximum number of faults/events can be read in Instance 0, Attribute 2.

Instances		Device
(Hex.)	(Dec.)	
0x0000…0x3FFF	0…16383	Host Drive
0x4000…0x43FF	16384…17407	Option Module
0x4400…0x47FF	17408…18431	Port 1
0x4800…0x4BFF	18432…19455	Port 2
0x4C00…0x4FFF	19456…20479	Port 3
0x5000…0x53FF	20480…21503	Port 4
0x5400…0x57FF	21504…22527	Port 5
0x5800…0x5BFF	22528…23551	Port 6
0x5C00…0x5FFF	23552…24575	Port 7
0x6000…0x63FF	24576…25599	Port 8
0x6400…0x67FF	25600…26623	Port 9
0x6800…0x6BFF	26624…27647	Port 10
0x6C00…0x6FFF	27648…28671	Port 11
0x7000…0x73FF	28672…29695	Port 12
0x7400…0x77FF	29696…30719	Port 13
0x7800…0x7BFF	30720…31743	Port 14

Example	Description
0	Class Attributes (Drive)
1	Most Recent Drive Fault
2	Second Most Recent Drive Fault
⋮	⋮
16384	Class Attributes (Option Module)
16385	Most Recent Option Module Event
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	UINT	Revision of object
2	Get	Number of Instances	UINT	Maximum number of faults/events that the device can record in its queue
3	Set	Fault Command Write	USINT	0 = No Operation 1 = Clear Fault/Event 2 = Clear Fault/Event Queue 3 = Reset Device
4	Get	Fault Trip Instance Read	UINT	Fault that tripped the device. For Option Modules, this value is always 1 when faulted.
5	Get	Fault Data List	STRUCT of: USINT USINT UINT[n]	Reserved
6	Get	Number of Recorded Faults	UINT	Number of faults/events in the queue. A '0' indicates the fault queue is empty.
7	Get	Fault Parameter Reference	UINT	Reserved

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Full/All Information	STRUCT of UINT STRUCT of: USINT USINT STRING[16] STRUCT of: LWORD BOOL[16] UINT CONTAINER[n]	Fault code Fault source DPI port DPI Device Object Fault text Fault time stamp Timer value (0 = timer not supported) BOOL[0]: (0 = invalid data, 1 = valid data) BOOL[1]: (0 = elapsed time, 1 = real time) BOOL[2…15]: Not used Reserved Reserved
1	Get	Basic Information	STRUCT of UINT STRUCT of: USINT USINT STRUCT of: LWORD BOOL[16]	Fault code Fault source DPI port DPI Device Object Fault time stamp Timer value (0 = timer not supported) BOOL[0]: (0 = invalid data, 1 = valid data) BOOL[1]: (0 = elapsed time, 1 = real time) BOOL[2…15]: Not used
2	Get	International Fault Text	STRINGN	Text describing the fault with support for Unicode.

DPI Alarm Object

Class Code

Hexadecimal	Decimal
0x98	152

Products such as PowerFlex drives use this object for alarms or warnings. Option Modules do not support this object.

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the maximum number of alarms supported by the queue. The maximum number of alarms can be read in Instance 0, Attribute 2.

Instances		Device
(Hex.)	(Dec.)	
0x0000…0x3FFF	0…16383	Host Drive

Only host devices can have alarms.

Example	Description
0	Class Attributes (Drive)
1	Most Recent Alarm
2	Second Most Recent Alarm
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	UINT	Revision of object
2	Get	Number of Instances	UINT	Maximum number of alarms that the device can record in its queue
3	Set	Alarm Command Write	USINT	0 = No Operation 1 = Clear Alarm 2 = Clear Alarm Queue 3 = Reset Device
4	Get	Fault Data List	STRUCT of: USINT USINT UINT[n]	Reserved
5	Get	Number of Recorded Alarms	UINT	Number of alarms in the queue. A '0' indicates the alarm queue is empty.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Full/All Information	STRUCT of UINT	Alarm code
			STRUCT of:	Alarm source
			USINT	DPI port
			USINT	DPI Device Object
			STRING[16]	Alarm text
			STRUCT of:	Alarm time stamp
			LWORD	Timer value (0 = timer not supported)
			BOOL[16]	BOOL[0]: (0 = invalid data, 1 = valid data)
				BOOL[1]: (0 = elapsed time, 1 = real time)
				BOOL[2…15] Reserved
			UINT	Reserved
			CONTAINER[n]	Reserved
1	Get	Basic Information	STRUCT of UINT	Alarm code
			STRUCT of:	Alarm source
			USINT	DPI port
			USINT	DPI Device Object
			STRUCT of:	Alarm time stamp
			LWORD	Timer value (0 = timer not supported)
			BOOL[16]	BOOL[0]: (0 = invalid data, 1 = valid data)
				BOOL[1]: (0 = elapsed time, 1 = real time)
				BOOL[2…15] Reserved
2	Get	International Alarm Text	STRINGN	Text describing the alarm with support for Unicode.

DPI Diagnostic Object

Class Code

Hexadecimal	Decimal
0x99	153

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the maximum number of diagnostic items in the device. The total number of diagnostic items can be read in Instance 0, Attribute 2.

Instances		Device
(Hex.)	(Dec.)	
0x0000…0x3FFF	0…16383	Host Drive
0x4000…0x43FF	16384…17407	Option Module
0x4400…0x47FF	17408…18431	Port 1
0x4800…0x4BFF	18432…19455	Port 2
0x4C00…0x4FFF	19456…20479	Port 3
0x5000…0x53FF	20480…21503	Port 4
0x5400…0x57FF	21504…22527	Port 5
0x5800…0x5BFF	22528…23551	Port 6
0x5C00…0x5FFF	23552…24575	Port 7
0x6000…0x63FF	24576…25599	Port 8
0x6400…0x67FF	25600…26623	Port 9
0x6800…0x6BFF	26624…27647	Port 10
0x6C00…0x6FFF	27648…28671	Port 11
0x7000…0x73FF	28672…29695	Port 12
0x7400…0x77FF	29696…30719	Port 13
0x7800…0x7BFF	30720…31743	Port 14

Example	Description
0	Class Attributes (Drive)
1	Drive Diagnostic Item 1
2	Drive Diagnostic Item 2
⋮	⋮
16384	Class Attributes (Option Module)
16385	Option Module Diagnostic Item 1
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	UINT	1
2	Get	Number of Instances	UINT	Number of diagnostic items in the device
3	Get	ENUM Offset	UINT	DPI ENUM object instance offset

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Full/All Information	STRUCT of:	
			BOOL[32]	Descriptor (see page 97)
			CONTAINER (1)	Value
			CONTAINER	Minimum value
			CONTAINER	Maximum value
			CONTAINER	Default value
			UINT	Pad Word
			UINT	Pad Word
			STRING[4]	Units (for example, Amps, Hz)
			UINT	Multiplier (2)
			UINT	Divisor (2)
			UINT	Base (2)
			INT	Offset (2)
			UDINT	Link (source of the value) (0 = no link)
			STRING[16]	Diagnostic name text
1	Get/Set	Value	Various	Diagnostic item value
2	Get	International Diagnostic Item Text	Struct of:	
			STRINGN	Diagnostic name text
			STRINGN	Diagnostic units text
3	Get	International Full Read All	STRUCT of:	
			BOOL[32]	Descriptor
			CONTAINER	Value
			CONTAINER	Minimum
			CONTAINER	Maximum
			CONTAINER	Default
			UINT	Pad word
			UINT	Pad word
			UINT	Multiplier
			UINT	Divisor
			UINT	Base
			INT	Offset
			UDINT	Pad
			BOOL[32]	Extended descriptor
			STRINGN	Diagnostic name text
			STRINGN	Diagnostic units text

(1) A CONTAINER is a 32-bit block of data that contains the data type used by a value. If signed, the value is sign extended. Padding is used in the CONTAINER to be sure that it is always 32-bits.

(2) This value is used in the formulas used to convert the value between display units and internal units. See Formulas for Converting on page 99.

DPI Time Object

Class Code

Hexadecimal	Decimal
0x9B	155

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Instances

The number of instances depends on the number of timers in the device. Instance 1 is always reserved for a real time clock although a device may not support it. The total number of timers can be read in Instance 0, Attribute 2.

Instances		Device
(Hex.)	(Dec.)	
0x0000…0x3FFF	0…16383	Host Drive
0x4000…0x43FF	16384…17407	Option Module
0x4400…0x47FF	17408…18431	Port 1
0x4800…0x4BFF	18432…19455	Port 2
0x4C00…0x4FFF	19456…20479	Port 3
0x5000…0x53FF	20480…21503	Port 4
0x5400…0x57FF	21504…22527	Port 5
0x5800…0x5BFF	22528…23551	Port 6
0x5C00…0x5FFF	23552…24575	Port 7
0x6000…0x63FF	24576…25599	Port 8
0x6400…0x67FF	25600…26623	Port 9
0x6800…0x6BFF	26624…27647	Port 10
0x6C00…0x6FFF	27648…28671	Port 11
0x7000…0x73FF	28672…29695	Port 12
0x7400…0x77FF	29696…30719	Port 13
0x7800…0x7BFF	30720…31743	Port 14

Example	Description
0	Class Attributes (Drive)
1	Real Time Clock (Predefined) (not always supported)
2	Timer 1
3	Timer 2
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Class Revision	UINT	Revision of object
2	Get	Number of Instances	UINT	Number of timers in the object, excluding the real time clock that is predefined.
3	Get	First Device Specific Timer	UINT	Instance of the first timer that is not predefined.
4	Set	Time Command Write	USINT	0 = No Operation 1 = Clear all timers (Does not clear real time clock or read only timers)
5	Get	Number of Supported Time Zones	UINT	Number of time zones described in the Time Zone List attribute.
6	Get	Time Zone List	STRUCT	Identifies a time zone.
7	Get/Set	Active Time Zone ID	UINT	The ID field of the Time Zone List structure for the desired time zone.

Attribute ID	Access Rule	Name	Data Type	Description
8	Get	Active Time Zone Data	Struct of:	
			INT	Standard bias
			USINT	Standard month
			USINT	Standard day of week
			USINT	Standard week
			USINT	Standard hour
			USINT	Standard minute
			USINT	Standard second
			INT	Daylight offset
			USINT	Daylight month
			USINT	Daylight day of week
			USINT	Daylight week
			USINT	Daylight hour
			USINT	Daylight minute
			USINT	Daylight second
9	Get/Set	Custom Time Zone Data	Struct of:	
			INT	Standard bias
			USINT	Standard month
			USINT	Standard day of week
			USINT	Standard week
			USINT	Standard hour
			USINT	Standard minute
			USINT	Standard second
			INT	Daylight offset
			USINT	Daylight month
			USINT	Daylight day of week
			USINT	Daylight week
			USINT	Daylight hour
			USINT	Daylight minute
			USINT	Daylight second

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Read Full	STRUCT of:	
			STRING[16]	Name of the timer
			LWORD or STRUCT	Elapsed time in milliseconds unless timer is a real time clock (see attribute 2)
			BOOL[16]	See Attribute 3
1	Get	Timer Text	STRING[16]	Name of the timer
2	Get/Set	Timer Value	LWORD -or- STRUCT of:	Elapsed time in milliseconds unless the timer is a real time clock. Real Time Clock Data:
			UINT	Milliseconds (0…999)
			USINT	Seconds (0…59)
			USINT	Minutes (0…59)
			USINT	Hours (0…23)
			USINT	Days (1…31)
			USINT	Months (1 = January, 12 = December)
			USINT	Years (since 1972)
3	Get	Timer Descriptor	BOOL[16]	BOOL[0]: (0 = invalid data, 1 = valid data) BOOL[1]: (0 = elapsed time, 1 = real time) BOOL[2…15]: Not used
4	Get	International Read Full	Struct of:	
			STRINGN	International timer text
			STRUCT	Timer value
			BOOL[16]	Timer descriptor
5	Get	International Timer Text	STRINGN	Name of this timer
6	Get	Clock Status	BOOL[32]	Identifies clock status
8	Get/Set	Number of Leap Seconds	INT	Identifies the current number of Leap Seconds.
9	Get	Clock Options	BOOL[32]	Identifies the optional functionality available in the device's System Clock.
10	Get/Set	Clock Options Enable	BOOL[32]	Identifies which of the clock's options are enabled.

Host DPI Parameter Object

Class Code

Hexadecimal	Decimal
0x9F	159

To access 'Device' parameters, use the DPI Parameter Object (Class Code 0x93).

Instances

The number of instances depends on the number of parameters in the device. The total number of parameters can be read in Instance 0, Attribute 0.

Instances		Device
(Hex.)	(Dec.)	
0x0000…0x3FFF	0…16383	Reserved
0x4000…0x43FF	16384…17407	Option Module
0x4400…0x47FF	17408…18431	Port 1
0x4800…0x4BFF	18432…19455	Port 2
0x4C00…0x4FFF	19456…20479	Port 3
0x5000…0x53FF	20480…21503	Port 4
0x5400…0x57FF	21504…22527	Port 5
0x5800…0x5BFF	22528…23551	Port 6
0x5C00…0x5FFF	23552…24575	Port 7
0x6000…0x63FF	24576…25599	Port 8
0x6400…0x67FF	25600…26623	Port 9
0x6800…0x6BFF	26624…27647	Port 10
0x6C00…0x6FFF	27648…28671	Port 11
0x7000…0x73FF	28672…29695	Port 12
0x7400…0x77FF	29696…30719	Port 13
0x7800…0x7BFF	30720…31743	Port 14

Example	Description
16384	Class Attributes (Option Module)
16385	Option Module Parameter 1 Attributes
16386	Option Module Parameter 2 Attributes
⋮	⋮
17408	Class Attributes (HIM)
17409	HIM Parameter 1 Attributes
17410	HIM Parameter 2 Attributes
⋮	⋮

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
0	Get	Number of Instances	UINT	Number of parameters in the device
1	Set	Write Protect Password	UINT	0 = Password disabled n = Password
2	Set	NVS Command Write	USINT	0 = No Operation 1 = Store values in active memory to NVS 2 = Load values in NVS to active memory 3 = Load default values to active memory
3	Get	NVS Parameter Value Checksum	UINT	Checksum of all parameter values in a user set in NVS
4	Get	NVS Link Value Checksum	UINT	Checksum of parameter links in a user set in NVS
5	Get	First Accessible Parameter	UINT	First parameter available if parameters are protected by passwords. A '0' indicates all parameters are protected.
7	Get	Class Revision	UINT	2 = DPI
8	Get	First Parameter Processing Error	UINT	The first parameter that has been written with a value outside of its range. A '0' indicates no errors.
9	Set	Link Command	USINT	0 = No Operation 1 = Clear All Parameter Links (This does not clear links to function blocks.)

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
6	Get	DPI Offline Read Full	STRUCT of:	
			BOOL[32]	Descriptor
			CONTAINER	Offline Minimum value
			CONTAINER	Offline Maximum value
			CONTAINER	Offline Default value
			STRING[16]	Parameter name
			STRING[4]	Offline parameter units
			UINT	Online minimum parameter instance
			UINT	Online maximum parameter instance
			UINT	Online default parameter instance
			UINT	Multiplier parameter instance
			UINT	Divisor parameter instance
			UINT	Base parameter instance
			UINT	Offset parameter instance
			USINT	Formula number
			USINT	Pad byte (always zero)
			UINT	Help instance
			UINT	Pad word (always a value of zero)
			CONTAINER	Parameter value
			UINT	Multiplier
			UNIT	Divisor
			UNIT	Base
			INT	Offset
7	Get	DPI Online Read Full	STRUCT of:	
			BOOL[32]	Descriptor (see page 111)
			CONTAINER[1]	Parameter value
			CONTAINER	Minimum value
			CONTAINER	Maximum value
			CONTAINER	Default value
			UINT	Next parameter
			UINT	Previous parameter
			STRING[4]	Units (for example, Amps, Hz)
			UINT	Multiplier [2]
			UINT	Divisor [2]
			UINT	Base [2]
			INT	Offset [2]
			USINT[3]	Link (source of the value) (0 = no link)
			USINT	Always zero (0)
			STRING[16]	Parameter name
8	Get	DPI Descriptor	BOOL[32]	Descriptor (see page 111)
9	Get/Set	DPI Parameter Value	Various	Parameter value in NVS. [3]
10	Get/Set	DPI RAM Parameter Value	Various	Parameter value in temporary memory. Valid only for DPI drives.
11	Get/Set	DPI Link	USINT[3]	Link (parameter or function block that is the source of the value) (0 = no link)
12	Get	Help Object Instance	UINT	ID for help text for this parameter
13	Get	DPI Read Basic	STRUCT of:	
			BOOL[32]	Descriptor (see page 111)
			CONTAINER	Parameter value
			CONTAINER	Minimum value
			CONTAINER	Maximum value
			CONTAINER	Default value
			STRING[16]	Parameter name
			STRING[4]	Units (for example, Amps, Hz)
14	Get	DPI Parameter Name	STRING[16]	Parameter name
15	Get	DPI Parameter Alias	STRING[16]	Customer supplied parameter name.

Attribute ID	Access Rule	Name	Data Type	Description
16	Get	Parameter Processing Error	USINT	0 = No error 1 = Value is less than the minimum 2 = Value is greater than the maximum
18	Get	International DPI Offline Parameter Text	Struct of: STRINGN STRINGN	 International parameter name International offline units
19	Get	International DPI Online Parameter Text	Struct of: STRINGN STRINGN	 International parameter name International online units
20	Get	International DPI Online Read Full	Struct of: BOOL[32] CONTAINER CONTAINER CONTAINER CONTAINER UINT UINT UINT UINT UINT INT USINT[3] USINT BOOL[32] STRINGN STRINGN	 Descriptor Parameter value Online minimum value Online maximum value Online default value Next Previous Multiplier Divisor Base Offset Link Pad word (always zero) Extended descriptor International parameter name International online parameter units
21	Get	DPI Extended Descriptor	UDINT	Extended Descriptor (see page 112)
22	Get	International DPI Offline Read Full	Struct of: BOOL CONTAINER CONTAINER CONTAINER UINT UINT UINT UINT UINT UINT USINT USINT UINT UINT CONTAINER UINT UINT UINT INT BOOL[32] STRINGN STRINGN	 Descriptor Offline minimum value Offline maximum value Offline default value Online minimum parameter instance Online maximum parameter instance Online default parameter instance Multiplier parameter instance Divisor parameter instance Base parameter instance Offset parameter instance Formula number Pad word (always zero) Help instance Pad word (always a value of zero) Parameter value Multiplier Divisor Base Offset Extended DPI descriptor International DPI parameter name International DPI offline parameter units

(1) A CONTAINER is a 32-bit block of data that contains the data type used by a parameter value. If signed, the value is sign extended. Padding is used in the CONTAINER to be sure that it is always 32-bits.

(2) This value is used in the formulas used to convert the parameter value between display units and internal units. See Formulas for Converting on page 113.

(3) Do NOT continually write parameter data to NVS. See the attention on page 49.

Descriptor Attributes

Bit	Name	Description
0	Data Type (Bit 1)	Right bit is least significant bit (0). 000 = USINT used as an array of Boolean 001 = UINT used as an array of Boolean 010 = USINT (8-bit integer) 011 = UINT (16-bit integer) 100 = UDINT (32-bit integer) 101 = TCHAR ((8-bit (not Unicode) or 16-bits (Unicode)) 110 = REAL (32-bit floating point value) 111 = Use bits 16, 17, 18
1	Data Type (Bit 2)	
2	Data Type (Bit 3)	
3	Sign Type	0 = unsigned 1 = signed
4	Hidden	0 = visible 1 = hidden
5	Not a Link Sink	0 = May be the sink end of a link 1 = May not be the sink end of a link
6	Not Recallable	0 = Recallable from NVS 1 = Not Recallable from NVS
7	ENUM	0 = No ENUM text 1 = ENUM text
8	Writable	0 = Read only 1 = Read/write
9	Not Writable When Enabled	0 = Writable when enabled (for example, drive running) 1 = Not writable when enabled
10	Instance	0 = Parameter value is not a Reference to another parameter 1 = Parameter value refers to another parameter
11	Uses Bit ENUM Mask	This parameter instance supports the Bit ENUM Mask attribute. For more information, see the definition of the attribute.
12	Decimal Place (Bit 0)	Number of digits to the right of the decimal point. 0000 = 0 1111 = 15
13	Decimal Place (Bit 1)	
14	Decimal Place (Bit 2)	
15	Decimal Place (Bit 3)	
16	Extended Data Type (Bit 4)	Bit 16 is the least significant bit. 000 = Reserved 001 = UDINT used as an array of Boolean 010 = Reserved 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved
17	Extended Data Type (Bit 5)	
18	Extended Data Type (Bit 6)	
19	Parameter Exists	Used to mark parameters that are not available to network tools.
20	Not Used	Reserved
21	Formula Links	Indicates the Formula Data is derived from other parameters.
22	Access Level (Bit 1)	A 3-bit field used to control access to parameter data.
23	Access Level (Bit 2)	
24	Access Level (Bit 3)	
25	Writable ENUM	ENUM text: 0 = Read Only, 1 = Read/Write
26	Not a Link Source	0 = May be the source end of a link 1 = May not be the source end of a link
27	Enhanced Bit ENUM	Parameter supports enhanced bit ENUMs.
28	Enhanced ENUM	Parameter supports enhanced ENUMs.
29	Uses DPI Limits Object	Parameter uses the DPI Limits Object. Intelligent offline tools make use of the Limits Object to select limits and units.
30	Extended Descriptor	Parameter uses Extended Descriptor bits, which can be obtained by reading the DPI Extended Descriptor attribute for this parameter.
31	Always Upload/Download	Parameter shall always be included in uploads and downloads.

Extended Descriptor Attributes

Bit	Name	Description
0	Indirect Mode	0 = Analog (selects entire parameters) 1 = Digital (selects individual bits within parameters)
1	Indirect Type 0	Analog input list (Instance 0xFFFF)
2	Indirect Type 1	Digital input list (Instance 0xFFFE)
3	Indirect Type 2	Feedback list (Instance 0xFFFD)
4	Indirect Type 3	Analog output list (Instance 0xFFFC)
5	Indirect Type 4	Digital output list (Instance 0xFFFB)
6	Indirect Type 5	Undefined (Instance 0xFFFA)
7	Indirect Type 6	Undefined (Instance 0xFFF9)
8	Indirect Type 7	Undefined (Instance 0xFFF8)
9	Indirect Type 8	Undefined (Instance 0xFFF7)
10	Indirect Type 9	Undefined (Instance 0xFFF6)
11	Indirect Type 10	Undefined (Instance 0xFFF5)
12	Indirect Type 11	Undefined (Instance 0xFFF4)
13	Indirect Type 12	Undefined (Instance 0xFFF3)
14	Indirect Type 13	Undefined (Instance 0xFFF2)
15	Indirect Type 14	Parameter-specific list
16	FP Max Decimals Bit 0	These four bits are used on REAL parameters only. They indicate the maximum number of decimal places to be displayed for small values. A value of 0 indicates to not limit the number of decimal places used.
17	FP Max Decimals Bit 1	
18	FP Max Decimals Bit 2	
19	FP Max Decimals Bit 1	
20	Extended Parameter Reference	0 = Not an Extended Parameter Reference 1 = Extended Parameter Reference An Extended Parameter Reference contains a reference to another parameter. The value is formatted the same as an analog mode Indirect Selector parameter (SSpppp, where SS = slot number of device to which this Extended Parameter Reference is pointing, and pppp = number of the parameter or diagnostic item to which this Extended Parameter Reference is pointing). Note that an Extended Parameter Reference can only select parameters unlike an Indirect Selector. An Extended Parameter Reference could be used to configure a Datalink or show the source of a Reference (among other uses).
21	Uses Rating Table Object	This parameter has rating-dependent defaults and limits that can be obtained from the Rating Table Object. The Offline Read Full will include the default value for the smallest rating and limits that will accommodate the full range of values allowed in the family of devices using this particular combination of Family Code and Config Code. The Online Read Full will include the rating-dependent default and limit values for this particular combination of Family Code, Config Code, and Rating Code.
22	Writable Referenced Parameter	This bit must be zero unless the parameter is an Extended Parameter Reference. If the parameter is an Extended Parameter Reference, then: 0 = The referenced parameter may be read-only or writable. 1 = The referenced parameter must always be writable (including while running).
23	Disallow Zero	This bit must be zero unless the parameter is an Indirect Selector or Extended Parameter Reference. If the parameter is an Indirect Selector or Extended Parameter Reference, then: 0 = Allow zero 1 = Disallow zero If this bit is cleared (indicating that a value of zero is allowed), the device must support the 'Zero Text' parameter attribute so that a software tool or HIM can obtain text from the Zero Text parameter attribute. If this bit is set (indicating that a value of zero is disallowed), a software tool or HIM will not allow the user to enter a value of zero.
24	Datalink Out	This bit is used by offline tools and indicates that this is a Datalink Out parameter. Bit 20 must also be set.
25	Datalink In	This bit is used by offline tools and indicates that this is a Datalink In parameter. Bits 20 and 22 must also be set.
26	Not Writable While IO Active	This parameter cannot be written if the I/O data being exchanged between the Host and the peripheral is valid.
27	Command Parameter	This parameter commands the drive to take an action, such as 'Reset Defaults' or 'Autotune', and then returns to a value of zero. Offline software tools will not allow setting this parameter to anything other than a value of zero. If an offline file contains a Command Parameter with a non-zero value, the offline software tool will change the value to zero. Note that command parameters cannot have values that do not return to zero.

Bit	Name	Description
28	Current Value Is Default	This bit identifies a parameter that will not change if a 'Reset Defaults' is commanded. For example, if a drive contains a Language parameter that is set to German, setting defaults will leave the parameter set to German. Likewise, if the parameter is set to French, setting defaults will leave the parameter set to French.
29	Use Zero Text	If the 'Disallow Zero' bit is set, this bit must be cleared. If the 'Disallow Zero' bit is cleared, then: 0 = Use Disabled Text parameter class attribute. 1 = Use Zero Text parameter instance attribute.
30-31	Reserved	Reserved

Formulas for Converting

Display Value = ((Internal Value + Offset) x Multiplier x Base) / (Divisor x $10^{\text{Decimal Places}}$))

Internal Value = ((Display Value x Divisor x $10^{\text{Decimal Places}}$) / (Multiplier x Base)) - Offset

Common Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	Yes	Yes	Set_Attribute_Single

Object Specific Services

Service Code	Implemented for:		Service Name	Allocation Size (in bytes)	
	Class	Instance		Par. Number	Par. Value
0x4D	Yes	No	Get_Attributes_Scattered	4	4
0x4E	Yes	No	Set_Attributes_Scattered	4	4

The table below lists the parameters for the Get_Attributes_Scattered and Set_Attributes_Scattered object-specific service:

Name	Data Type	Description
Parameter Number	UDINT	Parameter to read or write
Parameter Value	UDINT	Parameter value write (zero when reading)

The response data appears in the following format:

Name	Data Type	Description
Parameter Number	UDINT	Parameter read or write (1)
Parameter Value	UDINT	Parameter value read (zero when writing) (2)

(1) If an error occurred, bit 15 will be turned on in the response.

(2) If an error occurred, the error code will appear instead of the value.

TCP/IP Interface Object

Class Code

Hexadecimal	Decimal
0xF5	245

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Instances

The adapter supports one instance of the TCP/IP Interface object.

Number	Description
0	Class Attributes
1	Object Attributes

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	The revision of this object

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Status of TCP/IP Network Interface	UDINT	Bit | Value 0…3 = Configuration status 0 = Not configured 1 = Configured using BOOTP, DHCP, or parameters 2 = Configured using Node Address switches 3…15 = Reserved 4 = Reserved 5 = Configuration pending 6 = Duplicate IP address 7…15 = Reserved

Attribute ID	Access Rule	Name	Data Type	Description
2	Get	Configuration Capability	UDINT	Bit | Value (0 = False, 1 = True) 0 = Supports BOOTP 1 = DNS Client (able to resolve host names by query to DNS server) 2 = DHCP Client (able to obtain network configuration through DHCP) 3 = DHCP-DNS Update (able to send its host name in the DHCP request) 4 = Configuration Settable (able to set the network configuration via TCP/IP object) 5 = Hardware Configurable (able to set the network configuration via the Node Address switches) 6 = Configuration change requires reset 7 = Address Conflict Detection (ACD) capable 8…31 = Reserved
3	Set	Configuration Control	UDINT	Bit | Value 0…3 = Startup configuration 0 = Use configuration saved in NVS 1 = Obtain configuration via BOOTP 2 = Obtain configuration via DHCP 3…15 = Reserved 4 = DNS Enabled (resolves host names by query to DNS server) 5…31 = Reserved
4	Get	Physical Link Object	STRUCT of: UINT Padded EPATH	 Path size Path
5	Get	Interface Configuration	STRUCT of: UDINT UDINT UDINT UDINT UDINT STRING	 Option module IP address Option module subnet mask Option module gateway address Primary name server Secondary name server Default domain name
6	Get	Host Name	STRING	Host name when using DHCP
10	Set	Select ACD	BOOL	Activates the use of ACD
11	Set	Last Conflict Detected	STRUCT of: USINT USINT[6] USINT[28]	 ACD Activity Remote MAC ARP PDU

Ethernet Link Object

Class Code

Hexadecimal	Decimal
0xF6	246

Services

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x4C	No	Yes	Get_and_Clear
0x10	No	Yes	Set_Attribute_Single

Instances

The adapter supports the following instance of the TCP/IP Interface object.

Number	Description
0	Class Attributes
1	ENET1 network port
2	ENET2 network port

Class Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Revision	UINT	The revision of this object

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description
1	Get	Interface Speed	UDINT	Speed in megabits per second (Mbs)
2	Get	Interface Flags	UDINT	Bit | Value 0 = Link status (0 = inactive, 1 = active) 1 = Duplex (0 = half duplex, 1 = full duplex) 2…31 = Reserved
3	Get	Physical Address	USINT[6]	MAC address (XX-XX-XX-XX-XX-XX) The first octet (USINT[0]) is on the left.
4	Get	Interface Counters	STRUCT of: UDINT UDINT UDINT UDINT UDINT UDINT UDINT UDINT UDINT UDINT UDINT	 Octets received Unicast packets received Non-unicast packets received Inbound packets received but discarded Inbound packets with errors (not discarded) Inbound packets with unknown protocol Octets sent Unicast packets sent Non-unicast packets sent Outbound packets discarded Outbound packets with errors

Attribute ID	Access Rule	Name	Data Type	Description
5	Get	Media Counters	STRUCT of:	RX = Received, TX = Transmitted
			UDINT	RX frames not having integral number of octets long
			UDINT	RX frames not passing FCS check
			UDINT	TX frames having one collision
			UDINT	TX frames having multiple collisions
			UDINT	Number of times of SQE test error message
			UDINT	TX Frames delayed first attempt by busy medium
			UDINT	Collisions detected later than 512 bit-times in trans.
			UDINT	TX frames failing due to excessive collisions
			UDINT	TX frames failing due to intern MAC sublayer TX error
			UDINT	Times of carrier sense condition loss during trans
			UDINT	RX frames exceeding the maximum frame size
			UDINT	RX frames failing due to intern MAC sublayer RX error
6	Set	Interface Control	STRUCT of:	
			WORD	Control bits
			UINT	Forced interface speed
7	Get	Interface Type	USINT	Type of interface; 2 = twisted-pair
10	Get	Interface Label	SHORT_STRING	'1' = ENET1 network port '2' = ENET2 network port

Notes:

Logic Command/Status Words: PowerFlex 750-Series Drives

This appendix presents the definitions of the Logic Command and Logic Status words that are used for PowerFlex 750-Series drives.

Logic Command Word

Logic Bits																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Command	Description
																															x	Normal Stop	0 = Not Normal Stop 1 = Normal Stop
																														x		Start (1)	0 = Not Start 1 = Start
																													x			Jog 1 (2)	0 = Not Jog 1 (Par. 556) 1 = Jog 1
																												x				Clear Fault (3)	0 = Not Clear Fault 1 = Clear Fault
																										x	x					Unipolar Direction	00 = No Command 01 = Forward Command 10 = Reverse Command 11 = Hold Direction Control
																									x							Manual	0 = Not Manual 1 = Manual
																								x								Reserved	
																						x	x									Accel Time	00 = No Command 01 = Use Accel Time 1 (Par. 535) 10 = Use Accel Time 2 (Par. 536) 11 = Use Present Time
																				x	x											Decel Time	00 = No Command 01 = Use Decel Time 1 (Par. 537) 10 = Use Decel Time 2 (Par. 538) 11 = Use Present Time
																			x													Ref Select 1	000 = No Command 001 = Ref A Select (Par. 545) 010 = Ref B Select (Par. 550) 011 = Preset 3 (Par. 573) 100 = Preset 4 (Par. 574) 101 = Preset 5 (Par. 575) 110 = Preset 6 (Par. 576) 111 = Preset 7 (Par. 577)
																		x														Ref Select 2	
																	x															Ref Select 3	
																x																Reserved	
															x																	Coast Stop	0 = Not Coast to Stop 1 = Coast to Stop
														x																		Current Limit Stop	0 = Not Current Limit Stop 1 = Current Limit Stop
													x																			Run (4)	0 = Not Run 1 = Run
												x																				Jog 2 (2)	0 = Not Jog 2 (Par. 557) 1 = Jog 2
											x																					Reserved	
										x																						Reserved	
									x																							Reserved	
								x																								Reserved	
							x																									Reserved	
						x																										Reserved	
					x																											Reserved	
				x																												Reserved	
			x																													Reserved	
		x																														Reserved	
	x																															Reserved	
x																																Reserved	

(1) A Not Stop condition (logic bit 0 = 0) must first be present before a 1 = Start condition will start the drive.
(2) A Not Stop condition (logic bit 0 = 0) must first be present before a 1 = Jog 1/Jog 2 condition will jog the drive. A transition to a '0' will stop the drive.
(3) To perform this command, the value must switch from '0' to '1'.
(4) A Not Stop condition (logic bit 0 = 0) must first be present before a 1 = Run condition will run the drive. A transition to a '0' will stop the drive.

Logic Status Word

Logic Bits																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Command	Description
																															x	Run Ready	0 = Not Ready to Run 1 = Ready to Run
																														x		Active	0 = Not Active 1 = Active
																													x			Command Direction	0 = Reverse 1 = Forward
																												x				Actual Direction	0 = Reverse 1 = Forward
																											x					Accelerating	0 = Not Accelerating 1 = Accelerating
																										x						Decelerating	0 = Not Decelerating 1 = Decelerating
																									x							Alarm	0 = No Alarm (Par. 959 & 960) 1 = Alarm
																								x								Fault	0 = No Fault (Par. 952 & 953) 1 = Fault
																							x									At Setpt Spd	0 = Not at Setpoint Speed 1 = At Setpoint Speed
																						x										Manual	0 = Manual Mode Not Active 1 = Manual Mode Active
																					x											Spd Ref ID 0	00000 = Reserved 00001 = Auto Ref A (Par. 545) 00010 = Auto Ref B (Par. 550) 00011 = Auto Preset Speed 3 (Par. 573) 00100 = Auto Preset Speed 4 (Par. 574) 00101 = Auto Preset Speed 5 (Par. 575) 00110 = Auto Preset Speed 6 (Par. 576) 00111 = Auto Preset Speed 7 (Par. 577) 01000 = Reserved 01001 = Reserved 01010 = Reserved 01011 = Reserved 01100 = Reserved 01101 = Reserved 01110 = Reserved 01111 = Reserved 10000 = Man Port 0 10001 = Man Port 1 10010 = Man Port 2 10011 = Man Port 3 10100 = Man Port 4 10101 = Man Port 5 10110 = Man Port 6 10111 = Reserved 11000 = Reserved 11001 = Reserved 11010 = Reserved 11011 = Reserved 11100 = Reserved 11101 = Man Port 13 (Embedded ENET) 11110 = Man Port 14 (Drive Logix) 11111 = Alternate Man Ref Sel
																				x												Spd Ref ID 1	
																			x													Spd Ref ID 2	
																		x														Spd Ref ID 3	
																	x															Spd Ref ID 4	
																x																Reserved	
															x																	Running	0 = Not Running 1 = Running
														x																		Jogging	0 = Not Jogging (Par. 556 & 557) 1 = Jogging
													x																			Stopping	0 = Not Stopping 1 = Stopping
												x																				DC Brake	0 = Not DC Brake 1 = DC Brake
											x																					DB Active	0 = Not Dynamic Brake Active 1 = Dynamic Brake Active
										x																						Speed Mode	0 = Not Speed Mode (Par. 309) 1 = Speed Mode
									x																							Position Mode	0 = Not Position Mode (Par. 309) 1 = Position Mode
								x																								Torque Mode	0 = Not Torque Mode (Par. 309) 1 = Torque Mode
							x																									At Zero Speed	0 = Not at Zero Speed 1 =At Zero Speed
						x																										At Home	0 = Not at Home 1 = At Home
					x																											At Limit	0 = Not at Limit 1 = At Limit
				x																												Current Limit	0 = Not at Current Limit 1 = At Current Limit
			x																													Bus Freq Reg	0 = Not Bus Freq Reg 1 = Bus Freq Reg
		x																														Enable On	0 = Not Enable On 1 = Enable On
	x																															Motor Overload	0 = Not Motor Overload 1 = Motor Overload
x																																Regen	0 = Not Regen 1= Regen

The following terms and abbreviations are used throughout this manual. For definitions of terms not listed here, see the Allen-Bradley Industrial Automation Glossary, publication AG-7.1.

Bridge A network device that can route messages from one network to another. A bridge also refers to a communication module in a ControlLogix controller that connects the controller to a network. See also scanner.

CIP (Common Industrial Protocol) CIP is the transport and application layer protocol used for messaging over EtherNet/IP, ControlNet, and DeviceNet networks. The protocol is used for implicit messaging (real-time I/O) and explicit messaging (configuration, data collection, and diagnostics).

Controller A controller, also called programmable logic controller, is a solid-state control system that has a user-programmable memory for storage of instructions to implement specific functions such as I/O control, logic, timing, counting, report generation, communication, arithmetic, and data file manipulation. A controller consists of a central processor, input/output interface, and memory. See also Scanner.

Datalinks A Datalink is a type of pointer used by PowerFlex 750-Series drives to transfer data to and from the controller. Datalinks allow specified parameter values to be accessed or changed without using explicit messages. When active, each 32-bit Datalink in a PowerFlex 750-Series drive consumes 4 bytes in the input image table and/or 4 bytes in the output image table of the controller.

Device-level Ring (DLR) An Ethernet topology that consist of multiple devices configured in a circle-style connection, implemented at the device level, and with no additional switches required.

DriveExplorer Software A tool for monitoring and configuring Allen-Bradley products and network communication option modules. It can be run on computers running various Microsoft Windows operating systems. DriveExplorer software, version 6.xx or later, can be used to configure this option module and PowerFlex drive. Information about DriveExplorer software and a free lite version can be accessed at http://www.ab.com/drives/driveexplorer.

DriveTools SP Software A software suite designed for running on various Microsoft Windows operating systems. This software suite provides a family of tools, including DriveExecutive software, that you can use to program, monitor, control, troubleshoot, and maintain Allen-Bradley products. DriveTools SP can be used with PowerFlex drives. Information about DriveTools SP can be accessed at http://www.ab.com/drives/drivetools.

EDS (Electronic Data Sheet) Files Simple text files that are used by network configuration tools to describe products so that you can easily commission them on a network. EDS files describe a product device type and revision. EDS files for many Allen-Bradley products can be found at http://www.ab.com/networks/eds.

EtherNet/IP Network EtherNet/IP (Industrial Protocol) is an open producer-consumer communication network based on the Ethernet standard (IEEE 802.3), TCP/IP, UDP/IP, and CIP. Designed for industrial communication, both I/O and explicit messages can be transmitted over the network. Each device is assigned a unique IP address and transmits data on the network. The number of devices that an EtherNet/IP network can support depends on the class of IP address. For example, a network with a Class C IP address can have 254 nodes.

General information about EtherNet/IP and the EtherNet/IP specification are maintained by the Open DeviceNet Vendor's Association (ODVA). ODVA is online at http://www.odva.org.

Explicit Messaging Explicit messages are used to transfer data that does not require continuous updates. They are typically used to configure, monitor, and diagnose devices over the network.

Gateway A device on a network that connects an individual network to a system of networks. When a node needs to communicate with a node on another network, a gateway transfers the data between the two networks. You need to configure the address for the gateway device in the option module if you want the option module to communicate with devices that are not on its network.

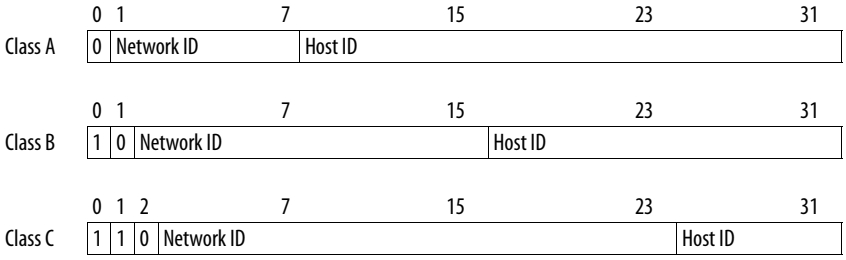
Hardware Address Each Ethernet device has a unique hardware address (sometimes called a MAC address) that is 48 bits. The address appears as six digits separated by colons (for example, xx:xx:xx:xx:xx:xx). Each digit has a value between 0 and 255 (0x00 and 0xFF). This address is assigned in the hardware and cannot be changed. It is required to identify the device if you are using a BOOTP/DHCP server.

I/O Data I/O data, sometimes called 'implicit messages' or 'input/output', is time-critical data such as a Logic Command and Reference. The terms 'input' (To Net) and 'output' (From Net) are defined from the controller's point of view. Output is produced by the controller and consumed by the network communication 20-750-ENETR option module or embedded EtherNet/IP adapter in a PowerFlex 755 drive. Input is produced by the option module or embedded adapter and consumed by the controller.

IP Addresses

A unique IP address identifies each node on an EtherNet/IP network. An IP address consists of 32 bits that are divided into four segments of one byte each. It appears as four decimal integers separated by periods (xxx.xxx.xxx.xxx). Each 'xxx' can have a decimal value from 0 to 255. For example, an IP address could be 192.168.0.1.

An IP address has two parts: a network ID and a host ID. The class of network determines the format of the address.



The number of devices on your EtherNet/IP network will vary depending on the number of bytes that are used for the network address. In many cases you are given a network with a Class C address, in which the first three bytes contain the network address (subnet mask = 255.255.255.0). This leaves 8 bits or 256 addresses on your network. Because two addresses are reserved for special uses (0 is an address for the network usually used by the router, and 255 is an address for broadcast messages to all network devices), you have 254 addresses to use on a Class C address block.

To be sure that each device on the Internet has a unique address, contact your network administrator or Internet Service Provider for unique fixed IP addresses. You can then set the unique IP address for the option module by using its rotary address switches, a DHCP or BOOTP server, or by manually configuring parameters in the option module. The option module reads the values of these parameters only at power-up.

Logic Command/Logic Status

The Logic Command is used to control the PowerFlex 750-Series drive (for example, start, stop, direction). It consists of one 32-bit word of output to the option module from the network. The definitions of the bits in this word are shown in Appendix B.

The Logic Status is used to monitor the PowerFlex 750-Series drive (for example, operating state, motor direction). It consists of one 32-bit word of input from the option module to the network. The definitions of the bits in this word are shown in Appendix B.

NVS (Nonvolatile Storage)

NVS is the permanent memory of a device. Devices such as the option module and drive store parameters and other information in NVS so that they are not lost when the device loses power. NVS is sometimes called 'EEPROM'.

Option Module

Devices such as drives, controllers, and computers usually require a network communication option module to provide a communication interface between them and a network such as EtherNet/IP. An option module reads data on the

network and transmits it to the connected device. It also reads data in the device and transmits it to the network.

The 20-750-ENETR Dual-port EtherNet/IP option module connects PowerFlex 750-Series drives to an EtherNet/IP network. Option modules are sometimes also called 'adapters', 'cards', 'embedded communication options', and 'peripherals'. On PowerFlex 750-Series drives, option modules can also be I/O modules, encoder modules, safety modules, and so forth.

PCCC (Programmable Controller Communications Command) PCCC is the protocol used by some controllers to communicate with devices on a network. Some software products (for example, DriveExplorer software and DriveExecutive software) also use PCCC to communicate.

Ping A message that is sent on the network to determine if a node exists.

PowerFlex 750-Series (Architecture Class) Drives Allen-Bradley PowerFlex 750-Series drives are part of the PowerFlex 7-Class family of drives.

Reference/Feedback The Reference is used to send a setpoint (for example, speed, frequency, torque) to the drive. It consists of one 32-bit word of output to the option module from the network.

Feedback is used to monitor the speed of the drive. It consists of one 32-bit word of input from the option module to the network.

RSLogix 5 Software and RSLogix 500 Software RSLogix software is a tool for configuring and monitoring controllers to communicate with connected devices. It is a 32-bit application that runs on various Windows operating systems. Information about RSLogix software can be found at http://www.software.rockwell.com/rslogix.

Scanner A scanner is a separate module (of a multi-module controller) or a built-in component (of a single-module controller) that provides communication with Option Modules connected to a network. See also Controller.

Subnet Mask An extension to the IP addressing scheme that lets you use a single network ID for multiple physical networks. A bit mask identifies the part of the address that specifies the network and the part of the address that specifies the unique node on the network. A '1' in the subnet mask indicates the bit is used to specify the network. A '0' in the subnet mask indicates that the bit is used to specify the node.

For example, a subnet mask on a network may appear as follows: 11111111 11111111 11111111 11000000 (255.255.255.192). This mask indicates that 26 bits are used to identify the network and 6 bits are used to identify devices on each network. Instead of a single physical Class C network with 254 devices, this subnet mask divides it into four networks with up to 62 devices each.

Switches Network devices that provide virtual connections that help to control collisions and reduce traffic on the network. They are able to reduce network congestion by transmitting packets to an individual port only if they are destined for the

connected device. In a control application, in which real time data access is critical, network switches may be required in place of hubs.

TCP (Transmission Control Protocol) EtherNet/IP uses this protocol to transfer Explicit Messaging packets using IP. TCP guarantees delivery of data through the use of retries.

UDP (User Datagram Protocol) EtherNet/IP uses this protocol to transfer I/O packets using IP. UDP provides a simple, but fast capability to send I/O messaging packets between devices. This protocol verifies that option modules transmit the most recent data because it does not use acknowledgements or retries.

UDDT (User-Defined Data Type) A structure data type that you define during the development of an application (for example, to convert 32-bit REAL parameter data for written and read values to correctly display them in human readable format).

Zero Data When communication is disrupted (for example, a cable is disconnected), the option module and drive can respond with zero data. Zero data results in the drive receiving zero as values for Logic Command, Reference, and Datalink data. If the drive was running and using the Reference from the option module, it will stay running but at zero Reference.

Notes:

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support/, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect^{SM} support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/support/.

Installation Assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/support/americas/phone_en.html, or contact your local Rockwell Automation representative.

New Product Satisfaction Return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 750COM-AT001A-EN-P - August 2012